

9.4

*Administration d' IBM MQ*

**IBM**

**Remarque**

Avant d'utiliser le présent document et le produit associé, prenez connaissance des informations générales figurant à la section [«Remarques»](#), à la page 595.

Cette édition s'applique à la version 9 édition 4 d' IBM® MQ et à toutes les éditions et modifications ultérieures, sauf indication contraire dans les nouvelles éditions.

Lorsque vous envoyez des informations à IBM, vous accordez à IBM le droit non exclusif d'utiliser ou de distribuer les informations de la manière qu'il juge appropriée, sans aucune obligation de votre part.

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

---

# Table des matières

<b>Administration.....</b>	<b>7</b>
Modes d'administration des gestionnaires de files d'attente IBM MQ et des ressources associées.....	8
Administration d' IBM MQ for Multiplatforms à l'aide de commandes de contrôle.....	11
Administration d' IBM MQ à l'aide de commandes MQSC.....	12
Syntaxe de la commande MQSC.....	13
Syntaxe du fichier d'entrée MQSC.....	16
Exécution interactive de commandes MQSC sous <b>runmqsc</b> .....	18
Exécution de commandes MQSC à partir de fichiers texte sous <b>runmqsc</b> .....	23
Configuration automatique à partir d'un script MQSC au démarrage.....	25
Automatisation de l'administration de IBM MQ à l'aide des commandes PCF.....	26
Introduction à IBM MQ Programmable Command Formats.....	27
Utilisation de MQAI pour simplifier l'utilisation des fichiers PCF.....	40
Administration à l'aide de REST API.....	76
Guide d'initiation à administrative REST API.....	77
Administration à distance à l'aide de REST API.....	82
REST API Horodatages.....	87
REST API traitement des erreurs.....	87
REST API découverte.....	90
REST APISupport de langue nationale (NLS).....	91
REST API Versions.....	92
Administration à l'aide de IBM MQ Console.....	94
Guide d'initiation à IBM MQ Console.....	94
Présentation rapide du IBM MQ Console.....	96
Paramètres IBM MQ Console.....	125
Administration à l'aide de IBM MQ Explorer.....	125
Ce que vous pouvez faire avec le IBM MQ Explorer.....	126
Configuration de IBM MQ Explorer.....	127
Utilisation de l'application IBM MQ Taskbar (Windows uniquement).....	133
L'application du moniteur d'alertes IBM MQ ( Windows uniquement).....	134
Utilisation des objets IBM MQ locaux.....	134
Utilisation des gestionnaires de files d'attente.....	135
Arrêt des canaux MQI.....	146
Utilisation des files d'attente locales.....	146
Utilisation des files d'attente éloignées.....	157
Utilisation de files d'attente alias.....	160
Utilisation des files d'attente modèles.....	162
Utilisation des files d'attente de rebut.....	163
Utilisation des rubriques d'administration.....	184
Utilisation des abonnements.....	187
Utilisation des services.....	192
Gestion des objets pour le déclenchement.....	200
Utilisation de l'utilitaire <b>dmpmqmsg</b> entre deux systèmes.....	202
Utilisation d'objets IBM MQ distants.....	206
Configuration des gestionnaires de files d'attente pour l'administration à distance.....	207
Gestion du serveur de commandes pour l'administration à distance.....	211
Emission de commandes MQSC sur un gestionnaire de files d'attente éloignées.....	212
Conversion de données entre des jeux de caractères codés.....	214
Administration de Managed File Transfer.....	219
Démarrage d'un agent MFT.....	220
Liste des agents MFT.....	226
Arrêt d'un agent MFT.....	226
Démarrage d'un nouveau transfert de fichiers.....	228

Création d'un transfert de fichiers planifié.....	231
Utilisation des transferts de fichiers en attente.....	233
Déclenchement d'un transfert de fichiers.....	233
Surveillance des transferts de fichiers en cours.....	235
Affichage du statut des transferts de fichiers dans le journal de transfert.....	237
Surveillance des ressources MFT.....	239
Utilisation des modèles de transfert de fichiers.....	272
Transfert de données de fichiers vers des messages.....	275
Transfert de données de messages vers des fichiers.....	291
Le pont de protocole.....	302
Le pont Connect:Direct.....	326
Utilisation de MFT à partir de IBM Integration Bus.....	342
MFT Reprise et redémarrage.....	343
Définition d'un délai d'attente pour la récupération des transferts bloqués.....	343
Administration de MQ Telemetry.....	349
Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Linux et AIX.....	350
Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Windows.....	352
Configuration de la mise en file d'attente répartie pour l'envoi de messages aux clients MQTT....	354
MQTT Identification, autorisation et authentification du client.....	356
Authentification de canal de télémétrie à l'aide de TLS.....	363
Confidentialité de la publication sur les canaux de télémétrie.....	367
Configuration TLS des clients MQTT Java et des canaux de télémétrie.....	368
Configuration JAAS du canal de télémétrie.....	373
Administration d'un client AMQP.....	375
Le service AMQP ne démarre pas automatiquement au démarrage du gestionnaire de files d'attente.....	375
Affichage des objets IBM MQ utilisés par les clients AMQP.....	376
Identification, autorisation et authentification du client AMQP.....	377
Confidentialité des publications sur les canaux.....	379
Configuration des clients AMQP avec TLS.....	380
Déconnexion des clients AMQP du gestionnaire de files d'attente.....	381
Administration de la multidiffusion.....	381
Initiation à la multidiffusion.....	381
Topologie de rubrique de multidiffusion IBM MQ.....	383
Contrôle de la taille des messages de multidiffusion.....	383
Activation de la conversion de données pour la messagerie multidiffusion.....	385
Surveillance des applications de multidiffusion.....	386
Fiabilité des messages de multidiffusion.....	387
Tâches de multidiffusion avancées.....	388
Administration de IBM MQ for IBM i.....	391
Gestion d'IBM MQ for IBM i à l'aide de commandes CL.....	391
Autres méthodes d'administration de IBM MQ for IBM i.....	406
Gestion des travaux pour IBM i.....	412
Disponibilité, sauvegarde, récupération et redémarrage sur IBM i.....	419
mise au repos IBM MQ for IBM i.....	465
Administering IBM MQ for z/OS.....	469
Issuing queue manager commands on z/OS.....	470
Using the operations and control panels on z/OS.....	484
Using the IBM MQ for z/OS utilities.....	492
Using the Command Facility on z/OS.....	495
Working with IBM MQ objects on z/OS.....	496
Implementing the system using multiple cluster transmission queues.....	498
Writing programs to administer IBM MQ for z/OS.....	501
Managing IBM MQ resources on z/OS.....	512
Recovery and restart on z/OS.....	550
IBM MQ and IMS.....	571
Operating Advanced Message Security on z/OS.....	583
Administration de IBM MQ Internet Pass-Thru.....	584

Démarrage et arrêt de MQIPT.....	584
administration de MQIPT via la ligne de commande.....	587
Exécution des sauvegardes.....	593
Optimisation des performances.....	594
<b>Remarques.....</b>	<b>595</b>
Documentation sur l'interface de programmation.....	596
Marques.....	596



# Administration de IBM MQ

---

Pour administrer vos gestionnaires de files d'attente IBM MQ et les ressources associées, choisissez la méthode de votre choix parmi un ensemble de tâches que vous pouvez utiliser pour activer et gérer ces ressources.

## Pourquoi et quand exécuter cette tâche

Vous pouvez administrer les objets IBM MQ localement ou à distance:

### administration locale

L'administration locale consiste à exécuter des tâches d'administration sur tous les gestionnaires de files d'attente que vous avez définis sur votre système local. Vous pouvez accéder à d'autres systèmes, par exemple via le programme d'émulation de terminal TCP/IP **telnet**, et y effectuer l'administration. Dans IBM MQ, vous pouvez considérer cela comme une administration locale car aucun canal n'est impliqué, c'est-à-dire que la communication est gérée par le système d'exploitation.

Pour plus d'informations, voir [«Utilisation des objets IBM MQ locaux»](#), à la page 134.

### Administration à distance


IBM MQ prend en charge l'administration à partir d'un point de contact unique via l'administration à distance. L'administration à distance vous permet d'émettre des commandes à partir de votre système local qui sont traitées sur un autre système et s'applique également à IBM MQ Explorer. Par exemple, vous pouvez émettre une commande distante pour modifier une définition de file d'attente sur un gestionnaire de files d'attente éloignées. Vous n'avez pas besoin de vous connecter à ce système, bien que les canaux appropriés doivent être définis. Le gestionnaire de files d'attente et le serveur de commandes sur le système cible doivent être en cours d'exécution.

Certaines commandes ne peuvent pas être émises de cette manière, notamment la création ou le démarrage de gestionnaires de files d'attente et de serveurs de commandes. Pour effectuer ce type de tâche, vous devez vous connecter au système distant et émettre les commandes à partir de celui-ci ou créer un processus qui peut émettre les commandes pour vous. Cette restriction s'applique également à IBM MQ Explorer.




Pour plus d'informations, voir [«Utilisation d'objets IBM MQ distants»](#), à la page 206.


Il existe un certain nombre de méthodes différentes que vous pouvez utiliser pour créer et administrer vos gestionnaires de files d'attente et leurs ressources associées dans IBM MQ. Ces méthodes incluent des interfaces de ligne de commande, des interfaces utilisateur graphiques et une API d'administration.

Il existe différents ensembles de commandes que vous pouvez utiliser pour administrer IBM MQ en fonction de votre plateforme:

- [«IBM MQ Les commandes de contrôle»](#), à la page 8
- [«Commandes IBM MQ Script \(MQSC\)»](#), à la page 8
- [«Formats PCF \(Programmable Command Formats\)»](#), à la page 9
- [administrative REST API](#)
-  [«Langage de contrôle IBM i \(CL\)»](#), à la page 10

Il existe également les autres options suivantes pour la création et la gestion des objets IBM MQ :

-   [«Le logiciel de gestion d'instance IBM MQ Explorer»](#), à la page 9
- [«Le logiciel de gestion d'instance IBM MQ Console»](#), à la page 9
-  [«Microsoft Cluster Service \(MSCS\)»](#), à la page 10

 Pour plus d'informations sur les interfaces et les options d'administration sous IBM MQ for z/OS, voir [«Administering IBM MQ for z/OS»](#), à la page 469.

Vous pouvez automatiser certaines tâches d'administration et de surveillance pour les gestionnaires de files d'attente locales et éloignées à l'aide des commandes PCF. Ces commandes peuvent également être simplifiées à l'aide de l'interface d'administration IBM MQ (MQAI) sur certaines plateformes. Pour plus d'informations sur l'automatisation des tâches d'administration, voir [«Automatisation de l'administration de IBM MQ à l'aide des commandes PCF»](#), à la page 26.

### Concepts associés

[Présentation technique du IBM MQ](#)

### Tâches associées

[Planification](#)

[Configuration](#)

### Référence associée

[Comparaison des ensembles de commandes](#)

## Modes d'administration des gestionnaires de files d'attente IBM MQ et des ressources associées

---

Vous pouvez administrer les gestionnaires de files d'attente IBM MQ et les ressources associées à l'aide des commandes de contrôle IBM MQ , IBM MQ Script Commands (MQSC), Programmable Command Formats (PCFs), administrative REST API, IBM MQ Console et IBM MQ Explorer. Pour IBM i , vous pouvez également utiliser le langage de contrôle IBM i et pour Windows , vous pouvez également utiliser le service de cluster Microsoft (MSCS).

### IBM MQ Les commandes de contrôle

#### Multi

Les commandes de contrôle permettent d'effectuer un certain nombre de tâches d'administration IBM MQ . Pour AIX, Linux®, and Windows, vous exécutez ces commandes sur la ligne de commande du système. Pour IBM i, vous exécutez ces commandes dans un Qshell. Voir [«Administration d' IBM MQ for Multiplatforms à l'aide de commandes de contrôle»](#), à la page 11.

### Commandes IBM MQ Script (MQSC)

Les commandes MQSC permettent de gérer les objets de gestionnaire de files d'attente, y compris le gestionnaire de files d'attente lui-même, les files d'attente, les définitions de processus, les listes de noms, les canaux, les canaux de connexion client, les programmes d'écoute, les services et les objets d'informations d'authentification.

#### ALW

Sous AIX, Linux, and Windows, vous ouvrez une invite de commande **runmqsc** , puis vous émettez des commandes MQSC vers un gestionnaire de files d'attente local ou distant à partir de cette invite. Vous pouvez effectuer cette opération de manière interactive ou exécuter une séquence de commandes à partir d'un fichier texte ASCII. Pour plus d'informations, voir [«Exécution interactive de commandes MQSC sous runmqsc»](#), à la page 18 et [«Exécution de commandes MQSC à partir de fichiers texte sous runmqsc»](#), à la page 23.

#### IBM i

Sous IBM i, vous créez une liste de commandes dans un fichier script, puis vous exécutez le fichier à l'aide de la commande **STRMQMMQSC** . Pour plus d'informations, voir [«Administration à l'aide de commandes MQSC sous IBM i»](#), à la page 407.

#### z/OS

Sous z/OS, les commandes MQSC peuvent être émises à partir de plusieurs sources, en fonction de la commande. Pour plus d'informations, voir [«Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS»](#), à la page 470.



## Formats PCF (Programmable Command Formats)

Les formats PCF (Programmable Command Formats) définissent les messages de commande et de réponse qui peuvent être échangés entre un programme et n'importe quel gestionnaire de files d'attente (qui prend en charge les PCF) d'un réseau. Vous pouvez utiliser des commandes PCF dans un programme d'application de gestion des systèmes pour l'administration des objets IBM MQ : objets d'informations d'authentification, canaux, programmes d'écoute de canal, listes de noms, définitions de processus, gestionnaires de files d'attente, files d'attente, services et classes de stockage. L'application peut fonctionner à partir d'un point unique du réseau pour communiquer des informations de commande et de réponse avec n'importe quel gestionnaire de files d'attente, local ou distant, à l'aide du gestionnaire de files d'attente local.

Pour plus d'informations sur les fichiers PCF, voir [«Introduction à IBM MQ Programmable Command Formats»](#), à la page 27.

Pour la définition des fichiers PCF et des structures des commandes et des réponses, voir [Programmable command formats reference](#).

## Le logiciel de gestion d'instance administrative REST API

administrative REST API fournit une interface RESTful que vous pouvez utiliser pour administrer IBM MQ. Lorsque vous utilisez administrative REST API, vous appelez une méthode HTTP sur une URL qui représente un objet IBM MQ . Par exemple, vous pouvez demander des informations sur les installations IBM MQ à l'aide de la méthode HTTP GET sur l'URL suivante:

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

Vous pouvez utiliser administrative REST API avec l'implémentation HTTP/REST d'un langage de programmation, ou à l'aide d'outils tels que cURL, ou d'un module complémentaire de navigateur client REST.

Pour plus d'informations, voir [administrative REST API](#)

## Le logiciel de gestion d'instance IBM MQ Console

Vous pouvez utiliser IBM MQ Console pour administrer IBM MQ à partir d'un navigateur Web.

Pour plus d'informations, voir [«Administration à l'aide de IBM MQ Console»](#), à la page 94.


## Le logiciel de gestion d'instance IBM MQ Explorer



À l'aide de l' IBM MQ Explorer, vous pouvez effectuer les actions suivantes:

- Définissez et contrôlez diverses ressources, telles que les gestionnaires de files d'attente, les files d'attente, les définitions de processus, les listes de noms, les canaux, les canaux de connexion client, les programmes d'écoute, les services et les clusters.
- Démarrez ou arrêtez un gestionnaire de files d'attente local et ses processus associés.
- Affichez les gestionnaires de files d'attente et les objets associés sur votre poste de travail ou à partir d'autres postes de travail.
- Vérifiez le statut des gestionnaires de files d'attente, des clusters et des canaux.
- Vérifiez les applications, les utilisateurs ou les canaux pour lesquels une file d'attente particulière est ouverte, à partir du statut de la file d'attente.

Sur les systèmes Windows et Linux for x86-64 , vous pouvez démarrer IBM MQ Explorer à l'aide du menu système ou du fichier exécutable MQExplorer .

 Sous Linux, pour démarrer correctement IBM MQ Explorer , vous devez pouvoir écrire un fichier dans votre répertoire de base et ce dernier doit exister.

Pour plus d'informations, voir [«Administration à l'aide de IBM MQ Explorer»](#), à la page 125.

Vous pouvez utiliser IBM MQ Explorer pour administrer les gestionnaires de files d'attente éloignées sur d'autres plateformes, y compris z/OS.

Depuis IBM MQ 9.3.0, IBM MQ Explorer a été supprimé du module d'installation d' IBM MQ . Il reste disponible sous forme de téléchargement séparé et peut être installé à partir du téléchargement autonome IBM MQ Explorer disponible à partir de Fix Central. Pour plus d'informations, voir [Installation et désinstallation de IBM MQ Explorer en tant qu'application autonome sous Linux et Windows](#).

## Langage de contrôle IBM i (CL)



Il s'agit de la méthode recommandée pour émettre des commandes d'administration à IBM MQ for IBM i. Les commandes peuvent être émises à partir de la ligne de commande ou en écrivant un programme CL. Ces commandes exécutent des fonctions similaires aux commandes PCF, mais le format est différent. Les commandes CL sont conçues exclusivement pour les serveurs et les réponses CL sont lisibles par l'utilisateur, alors que les commandes PCF sont indépendantes de la plateforme et que les formats de commande et de réponse sont destinés à être utilisés par les programmes.

Pour plus de détails sur le langage de contrôle IBM i (CL), voir [«Gestion d'IBM MQ for IBM i à l'aide de commandes CL»](#), à la page 391 et [Commandes CL IBM MQ for IBM i](#).

## Microsoft Cluster Service (MSCS)



Microsoft Cluster Service (MSCS) vous permet de connecter des serveurs à un *cluster*, ce qui offre une meilleure disponibilité des données et des applications et facilite la gestion du système. MSCS détecte automatiquement les échecs de serveurs ou d'applications et initie alors les restaurations requises.

Il est important de ne pas confondre les clusters au sens MSCS avec les clusters IBM MQ . La distinction est la suivante:

### Clusters IBM MQ

Il s'agit de groupes de deux ou plusieurs gestionnaires de files d'attente sur un ou plusieurs ordinateurs, fournissant une interconnexion automatique et permettant le partage de files d'attente entre eux pour l'équilibrage de charge et la redondance.

### Clusters MSCS

Il s'agit de groupes d'ordinateurs qui sont connectés entre eux et configurés de manière à ce que, en cas de défaillance, MSCS effectue une *reprise en ligne*, en transférant les données d'état des applications de l'ordinateur défaillant vers un autre ordinateur du cluster et en relayant leur fonctionnement.

La prise en charge de [Microsoft Cluster Service \(MSCS\)](#) fournit des informations détaillées sur la configuration de votre système IBM MQ for Windows pour l'utilisation de MSCS.

### Tâches associées

[«Administration d' IBM MQ à l'aide de commandes MQSC»](#), à la page 12

Vous pouvez utiliser les commandes MQSC pour gérer les objets de gestionnaire de files d'attente, notamment le gestionnaire de files d'attente lui-même, les files d'attente, les définitions de processus, les canaux, les canaux de connexion client, les programmes d'écoute, les services, les listes de noms, les clusters et les objets d'informations d'authentification. Les commandes MQSC sont disponibles sur toutes les plateformes.

### Référence associée

[Référence relative à l'administration](#)

# Administration d' IBM MQ for Multiplatforms à l'aide de commandes de contrôle

Les commandes de contrôle permettent d'effectuer un certain nombre de tâches d'administration IBM MQ . Pour AIX, Linux et Windows, vous exécutez ces commandes sur la ligne de commande du système. Pour IBM i, vous exécutez ces commandes dans un Qshell.

## Avant de commencer

Lorsque vous utilisez des commandes de contrôle qui fonctionnent sur un gestionnaire de files d'attente, vous devez utiliser la commande de l'installation associée au gestionnaire de files d'attente que vous utilisez.

Lors de l'utilisation de commandes de contrôle qui fonctionnent sur un gestionnaire de files d'attente configuré pour utiliser l'authentification de connexion avec CHCKLOCL (REQUIRED), et un échec de connexion est observé:

- Indiquez un ID utilisateur et un mot de passe si la commande de contrôle le permet.
- Utilisez les équivalents MQSC des commandes de contrôle lorsqu'elles existent.
- Démarrez le gestionnaire de files d'attente à l'aide de l'option -ns, alors que les commandes de contrôle qui ne peuvent pas se connecter doivent être exécutées.

**Remarque :** Différentes plateformes peuvent accepter des arguments de commande entrés dans un ordre différent. En particulier, cela signifie que les commandes qui fonctionnent sous Linux peuvent ne pas fonctionner sur d'autres plateformes. Pour cette raison, vous devez toujours entrer les arguments spécifiés dans les diagrammes de syntaxe.

Pour obtenir la liste complète des commandes de contrôle, voir [IBM MQ control commands reference](#).

## Procédure

- 

Utilisez les commandes de contrôle sur les systèmes AIX and Linux .

Dans les systèmes IBM MQ for AIX or Linux , vous entrez des commandes de contrôle dans une fenêtre shell.

Si vous souhaitez exécuter des commandes de contrôle, votre ID utilisateur doit être membre du groupe mqm pour la plupart des commandes de contrôle. Pour plus d'informations à ce sujet, voir [Droit d'administration d' IBM MQ sur AIX, Linux, and Windows](#). En outre, notez les informations spécifiques à l'environnement. pour la ou les plateformes utilisées par votre entreprise.

Dans les environnements UNIX and Linux , les commandes de contrôle, y compris le nom de la commande elle-même, les indicateurs et tous les arguments, sont sensibles à la casse. Par exemple, dans la commande:

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

- Le nom de la commande doit être crtmqmet non CRTMQM.
- L'indicateur doit être -u, et non -U.
- La file d'attente de rebut est appelée SYSTEM . DEAD . LETTER . QUEUE.
- L'argument est spécifié sous la forme jupiter . queue . manager, qui est différente de JUPITER . queue . manager.

Prenez soin de saisir les commandes exactement telles que vous les voyez dans les exemples.

- 

Utilisez les commandes de contrôle sur les systèmes Windows .

Dans IBM MQ for Windows, vous entrez des commandes de contrôle dans une invite de commande.

Si vous souhaitez exécuter des commandes de contrôle, votre ID utilisateur doit être membre du groupe mqm pour la plupart des commandes de contrôle. Pour plus d'informations à ce sujet, voir [Droit d'administration d' IBM MQ sur AIX, Linux, and Windows](#). En outre, notez les informations spécifiques à l'environnement pour la ou les plateformes utilisées par votre entreprise.

Les commandes de contrôle et leurs indicateurs ne sont pas sensibles à la casse, mais les arguments de ces commandes, tels que les noms de file d'attente et les noms de gestionnaire de files d'attente, sont sensibles à la casse.

Par exemple, dans la commande:

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

- Le nom de la commande peut être entré en majuscules ou en minuscules, ou un mélange des deux. Ces éléments sont tous valides: crtmqm, CRTMQMet CRTmqm.
- L'indicateur peut être entré sous la forme -u, -U, /uou /U.
- SYSTEM.DEAD.LETTER.QUEUE et jupiter.queue.manager doivent être entrés exactement comme indiqué.

#### IBM i

Utilisez les commandes de contrôle sur les systèmes IBM i .

Sous IBM MQ for IBM i, vous exécutez les commandes de contrôle à partir d'un environnement Qshell. Pour utiliser Qshell, entrez STRQSH sur la ligne de commande IBM i . Vous pouvez quitter la ligne de commande et y revenir à tout moment en appuyant sur F3.

Un petit nombre de commandes de contrôle ne sont pas prises en charge sous IBM i. Par exemple, les commandes d'installation multiple ne sont pas prises en charge car vous ne pouvez pas avoir plus d'une copie de IBM MQ sur un système IBM i . Les commandes qui ne sont pas prises en charge sous IBM i sont signalées **ALW** dans la [référence des commandes de contrôle IBM MQ](#).

#### Référence associée

[Informations de référence sur les commandes de contrôle IBM MQ](#)

## Administration d' IBM MQ à l'aide de commandes MQSC

Vous pouvez utiliser les commandes MQSC pour gérer les objets de gestionnaire de files d'attente, notamment le gestionnaire de files d'attente lui-même, les files d'attente, les définitions de processus, les canaux, les canaux de connexion client, les programmes d'écoute, les services, les listes de noms, les clusters et les objets d'informations d'authentification. Les commandes MQSC sont disponibles sur toutes les plateformes.

### Pourquoi et quand exécuter cette tâche

Les commandes MQSC disponibles sont détaillées dans la [référence des commandes MQSC](#).

La façon dont vous émettez les commandes MQSC dépend de votre plateforme:

- **ALW** Sous AIX, Linux, and Windows, vous émettez des commandes MQSC vers un gestionnaire de files d'attente à partir de l'invite de commande **runmqsc** . Vous pouvez utiliser cette invite de commande de plusieurs manières:
  - En mode interactif, émission de commandes MQSC à partir d'un clavier. Voir [«Exécution interactive de commandes MQSC sous runmqsc»](#), à la page 18.
  - Emission de commandes MQSC à partir d'un fichier texte ASCII. Voir [«Exécution de commandes MQSC à partir de fichiers texte sous runmqsc»](#), à la page 23.
  - Emission de commandes MQSC sur un gestionnaire de files d'attente éloignées. Voir [«Emission de commandes MQSC sur un gestionnaire de files d'attente éloignées»](#), à la page 212.

- **IBM i** Sous IBM i, vous créez une liste de commandes dans un fichier script, puis vous exécutez le fichier à l'aide de la commande **STRMQMMQSC** . Pour plus d'informations, voir [«Administration à l'aide de commandes MQSC sous IBM i»](#), à la page 407.
- **z/OS** Sous z/OS, les commandes MQSC peuvent être émises à partir de plusieurs sources, en fonction de la commande. Pour plus d'informations, voir [«Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS»](#), à la page 470.

## Procédure

- [«Syntaxe de la commande MQSC»](#), à la page 13
- [«MQSC: Caractères spéciaux et valeurs génériques»](#), à la page 15
- [«Exécution interactive de commandes MQSC sous runmqsc»](#), à la page 18
- [«Exécution de commandes MQSC à partir de fichiers texte sous runmqsc»](#), à la page 23
- [«Configuration automatique à partir d'un script MQSC au démarrage»](#), à la page 25

### Tâches associées

[Résolution des problèmes liés aux commandes MQSC](#)

### Référence associée

[runmqsc \(exécution des commandes MQSC\)](#)

## Syntaxe de la commande MQSC

Vous pouvez utiliser les commandes MQSC pour gérer les objets de gestionnaire de files d'attente. Les commandes MQSC sont disponibles sur toutes les plateformes. Certains éléments de la syntaxe de commande sont spécifiques à la plateforme.

### Séquence de paramètres

Chaque commande commence par un paramètre primaire (un verbe), suivi d'un paramètre secondaire (un nom). Il est ensuite suivi du nom ou du nom générique de l'objet (entre parenthèses) s'il en existe un, qui figure dans la plupart des commandes. Ensuite, les paramètres peuvent généralement se produire dans n'importe quel ordre ; si un paramètre a une valeur correspondante, la valeur doit se produire directement après le paramètre auquel il se rapporte.

**Remarque :** **z/OS** Sous z/OS, le paramètre secondaire n'a pas besoin d'être second.

### Blancs et virgules

Les mots clés, les parenthèses et les valeurs peuvent être séparés par un nombre quelconque de blancs et de virgules. Une virgule affichée dans les diagrammes de syntaxe peut toujours être remplacée par un ou plusieurs blancs. Au moins un blanc doit immédiatement précéder chaque paramètre (après le paramètre principal) sauf sur z/OS .

N'importe quel nombre de blancs peut apparaître au début ou à la fin de la commande et entre les paramètres, la ponctuation et les valeurs. Par exemple, la commande suivante est valide:

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

Les blancs à l'intérieur d'une paire de guillemets sont significatifs.


Des virgules supplémentaires peuvent apparaître partout où des blancs sont autorisés et sont traitées comme s'il s'agissait de blancs (à moins, bien entendu, qu'elles soient placées dans des chaînes entre guillemets).

## Paramètres répétés


Les paramètres répétés ne sont pas admis. La répétition d'un paramètre avec sa version "NO", comme dans REPLACE NOREPLACE, n'est pas non plus autorisée.

## Chaînes et apostrophes


Les chaînes contenant des blancs, des caractères minuscules ou des caractères spéciaux doivent être placées entre apostrophes, sauf si l'une des conditions suivantes est remplie:

- Les caractères spéciaux sont un ou plusieurs des caractères suivants:
  - le point (.)
  - la barre oblique (/)
  - le caractère de soulignement (\_)
  - le symbole de pourcentage (%)
-  La commande est émise à partir des panneaux d'opérations et de contrôle IBM MQ for z/OS.
- La chaîne est une valeur générique se terminant par un astérisque. (sous IBM i, ils doivent être placés entre apostrophes)
- La chaîne est un astérisque unique, par exemple, TRACE (\*) (sous IBM i, ils doivent être placés entre guillemets simples)
- La chaîne est une spécification de plage contenant un signe deux-points, par exemple, CLASS (01:03)

Si la chaîne elle-même contient une apostrophe, celle-ci est représentée par deux apostrophes.

 Sous Multiplateformes, une chaîne ne contenant aucun caractère (c'est-à-dire deux guillemets simples sans espace entre eux) est interprétée comme un espace vide entre guillemets simples, c'est-à-dire interprété de la même manière que ("). La seule exception à cette règle est si l'attribut utilisé est l'un des attributs suivants, lorsque deux guillemets simples sans espace sont interprétés comme une chaîne de longueur nulle:

- TOPICSTR
- SUB
- USERDATA
- SELECTOR

 Sous z/OS, si vous souhaitez placer un espace entre guillemets simples, vous devez l'entrer en tant que tel (" "). Une chaîne ne contenant aucun caractère ("") est identique à la chaîne entrée ().

Les blancs de fin des attributs de chaîne basés sur les types MQCHARV, tels que SELECTOR, les données de sous-utilisateur, sont traités comme significatifs, ce qui signifie que 'abc ' n'est pas égal à 'abc '.

## Parenthèses vides

Une parenthèse ouvrante suivie d'une parenthèse fermante, sans aucune information significative entre les deux, n'est pas valide sauf lorsqu'elle est spécifiquement indiquée. Par exemple, la chaîne suivante n'est pas valide:

```
NAME ( )
```

## Minuscules et majuscules

Les mots clés ne sont pas sensibles à la casse: AltER, alter et ALTER sont tous acceptables.

Tout ce qui n'est pas entre guillemets est plié en majuscules.

## Synonymes

Des synonymes sont définis pour certains paramètres. Par exemple, DEF est toujours un synonyme de DEFINE, de sorte que DEF QLOCAL est valide. Les synonymes ne sont cependant pas uniquement des chaînes minimales ; DEFI n'est pas un synonyme valide de DEFINE.

**Remarque :** Il n'existe pas de synonyme pour le paramètre DELETE. Cela permet d'éviter la suppression accidentelle d'objets lors de l'utilisation de DEF, synonyme de DEFINE.

## Caractères spéciaux

Les commandes MQSC utilisent certains caractères spéciaux pour avoir certaines significations. Pour plus d'informations sur ces caractères spéciaux et sur leur utilisation, voir «MQSC: Caractères spéciaux et valeurs génériques», à la page 15.

### Tâches associées

Résolution des problèmes liés aux commandes MQSC

### Référence associée

[runmqsc \(exécution des commandes MQSC\)](#)

## MQSC: Caractères spéciaux et valeurs génériques

Certains caractères, par exemple barre oblique inversée (\) et guillemet (") Les caractères ont des significations spéciales lorsqu'ils sont utilisés avec les commandes MQSC. Certains caractères spéciaux pouvant être utilisés avec des paramètres peuvent avoir des valeurs génériques mais doivent être spécifiés correctement.

Précède la barre oblique inversée (\) et le guillemet double (") caractères avec un \, c'est-à-dire, entrez \\ ou \" si vous souhaitez \ ou " dans votre texte.

Lorsqu'un paramètre peut avoir une valeur générique, il se termine par un astérisque (\*), par exemple ABC\*. Une valeur générique signifie toutes les valeurs commençant par ; par conséquent, ABC\* signifie toutes les valeurs commençant par ABC. Si des caractères qui requièrent des guillemets sont utilisés dans la valeur, l'astérisque doit être placé entre les guillemets, c'est-à-dire 'abc\*'. L'astérisque doit être le dernier ou le seul caractère de la valeur.

Le point d'interrogation (?) et le signe deux-points (:) ne sont pas autorisés dans les valeurs génériques.

Lorsque vous devez utiliser l'un de ces caractères spéciaux dans une zone (par exemple, dans le cadre d'une description), vous devez placer la chaîne entière entre guillemets simples.


Caractère	Description
	Les blancs sont utilisés comme séparateurs. Les blancs multiples sont équivalents à un blanc unique, sauf dans les chaînes placées entre apostrophes ('). Les blancs de fin de ces attributs de chaîne basés sur les types MQCHARV sont traités comme significatifs.
,	Les virgules sont utilisées comme séparateurs. Les virgules multiples sont équivalentes à une virgule unique, sauf dans les chaînes placées entre apostrophes (').
'	Une apostrophe indique le début ou la fin d'une chaîne. IBM MQ laisse tous les caractères placés entre guillemets exactement tels qu'ils sont entrés. Les apostrophes ne sont pas incluses lors du calcul de la longueur de la chaîne.
"	Les apostrophes à l'intérieur d'une chaîne sont traitées par IBM MQ comme un seul caractère lors du calcul de la longueur de la chaîne et la chaîne n'est pas terminée.
=	 Sur z/OS, un signe égal indique le début d'une valeur de paramètre qui se termine par une virgule ou un blanc.

Tableau 1. Description des caractères ayant des significations spéciales (suite)

Caractère	Description
(	Une parenthèse ouvrante indique le début d'une valeur de paramètre ou d'une liste de valeurs.
)	Une parenthèse fermante indique la fin d'une valeur de paramètre ou d'une liste de valeurs.
:	Un signe deux-points indique une plage inclusive. Par exemple (1: 5) signifie (1,2,3,4, 5). Cette notation ne peut être utilisée que dans les commandes <b>TRACE</b> .
*	Un astérisque signifie tout. Par exemple, <b>DISPLAY TRACE (*)</b> signifie l'affichage de toutes les traces et <b>DISPLAY QUEUE (PAY*)</b> signifie l'affichage de toutes les files d'attente dont le nom commence par PAY.




## Syntaxe du fichier d'entrée MQSC

Si vous disposez de commandes longues ou que vous utilisez une séquence de commandes particulière à plusieurs reprises, vous pouvez utiliser un fichier d'entrée pour émettre des commandes MQSC. Le contenu du fichier d'entrée doit respecter la syntaxe décrite dans cette rubrique.

### Présentation

Les commandes MQSC sont entrées via le *périphérique d'entrée standard*, également appelé `stdin`. Il s'agit généralement du clavier, mais vous pouvez spécifier que l'entrée doit provenir d'un fichier d'entrée.

Vous pouvez utiliser ce fichier d'entrée avec l'un des outils spécifiques à la plateforme suivants:

-  La commande **runmqsc** sous AIX, Linux, and Windows. Voir «[Exécution de commandes MQSC à partir de fichiers texte sous runmqsc](#)», à la page 23
-  La commande **STRMQM** sous IBM i. Voir «[Administration à l'aide de commandes MQSC sous IBM i](#)», à la page 407
-  Les fichiers d'initialisation CSQINP1, CSQINP2et CSQINPX ou l'utilitaire de traitement par lots CSQUTIL sous z/OS. Voir «[Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS](#)», à la page 470

### Syntaxe

Syntaxe du fichier d'entrée MQSC:

- Pour la portabilité entre les environnements IBM MQ , limitez la longueur de ligne dans les fichiers de commandes MQSC à 72 caractères.
- Chaque commande doit commencer sur une nouvelle ligne.
- Une ligne commençant par un astérisque (\*) en première position est ignorée. Vous pouvez l'utiliser pour insérer des commentaires dans le fichier.
- Les lignes vides sont ignorées.
- Un signe plus (+) indique que la commande se poursuit à partir du premier caractère non blanc de la ligne suivante. Si vous utilisez le signe + pour continuer une commande, n'oubliez pas de laisser au moins un blanc avant le paramètre suivant (sauf sous z/OS où cela n'est pas nécessaire). Les commentaires ou les lignes vides sont supprimés lorsque la commande est réassemblée en une seule chaîne.
- Un signe moins (-) indique que la commande doit être poursuivie à partir du début de la ligne suivante. Les commentaires ou les lignes vides sont supprimés lorsque la commande est réassemblée en une seule chaîne.
- Les commandes MQSC contenues dans une commande Escape PCF (Programmable Command Format) ne peuvent pas continuer avec le signe plus ou le signe moins. La totalité de la commande doit être



contenue dans une seule commande Escape. Pour plus d'informations sur les commandes PCF, voir «Introduction à IBM MQ Programmable Command Formats», à la page 27.

- Sous **Multiplateformes**, et sous z/OS pour les commandes émises à partir du programme utilitaire de traitement par lots CSQUTIL, vous pouvez utiliser un point-virgule (;) pour terminer une commande, même si vous avez entré un signe plus (+) à la fin de la ligne précédente.
- Une ligne ne doit pas se terminer par un caractère de contrôle du clavier (par exemple, une tabulation).
- Si vous exécutez la commande **runmqsc** en mode client en redirigeant stdin à partir d'un fichier texte et que vous indiquez l'indicateur **-u** pour fournir les données d'identification, la commande **runmqsc** ne demande pas de mot de passe et le mot de passe est lu à partir de stdin. Vous devez vous assurer que la première ligne de données fournie via stdin est le mot de passe. Pour ce faire, utilisez des outils de ligne de commande tels que "echo" ou "cat" et transmettez le mot de passe suivi du script MQSC à la **runmqsc** commande stdin.
- **Windows** Sous Windows, si certains caractères spéciaux tels que le signe dièse (£) et le caractère logique NOT (~) sont utilisés dans un script de commande (par exemple, dans le cadre d'une description d'objet), ils sont affichés différemment dans la sortie d'une commande telle que **DISPLAY QLOCAL**.

Voir aussi «Syntaxe de la commande MQSC», à la page 13.

## Exemples

L'exemple suivant est un extrait d'un fichier de commandes MQSC qui affiche la commande **DEFINE QLOCAL**.

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

Figure 1. Extrait à partir d'un fichier de commandes MQSC

Une fois la commande **runmqsc** exécutée, un rapport est renvoyé. L'exemple suivant est un extrait d'un rapport:

```

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

```

Figure 2. Extraction à partir d'un fichier de rapport de commandes MQSC

Vous pouvez également utiliser les exemples de fichiers de commandes MQSC pour vous aider à créer votre fichier texte:

#### **amqscos0.tst**

Définitions des objets utilisés par les exemples de programme.

#### **amqscic0.tst**

Définitions des files d'attente pour les transactions CICS .

**Linux** **AIX** Sous AIX and Linux, ces fichiers se trouvent dans le répertoire `MQ_INSTALLATION_PATH/samp`. `MQ_INSTALLATION_PATH` représente le répertoire de haut niveau dans lequel IBM MQ est installé.

**Windows** Sous Windows, ces fichiers se trouvent dans le répertoire `MQ_INSTALLATION_PATH\tools\mqsc\samples`. `MQ_INSTALLATION_PATH` représente le répertoire de haut niveau dans lequel IBM MQ est installé.

## **ALW** Exécution interactive de commandes MQSC sous `runmqsc`

Sous AIX, Linux, and Windows, vous pouvez utiliser l'invite de commande `runmqsc` pour émettre des commandes MQSC à un gestionnaire de files d'attente de manière interactive. L'exécution interactive est particulièrement adaptée aux tests rapides.

### **Avant de commencer**

Vous devez utiliser la commande `runmqsc` à partir de l'installation associée au gestionnaire de files d'attente que vous utilisez. Vous pouvez déterminer à quelle installation un gestionnaire de files d'attente est associé à l'aide de la commande `dspmqr -o installation`.

Vous pouvez plus facilement voir que vous êtes dans un environnement MQSC et voir certains détails de l'environnement en cours en définissant une invite de votre choix à l'aide de la variable d'environnement **MQPROMPT**. Pour plus d'informations, voir «Définition de l'invite de commande MQSC», à la page 21.

**Linux** **AIX** Lorsque vous exécutez des commandes MQSC de manière interactive sur des plateformes AIX and Linux, l'invite de commande `runmqsc` prend également en charge des fonctions d'éditeur de ligne de commande supplémentaires. Voir «Activation du rappel et de l'exécution des commandes et des touches de commande Emacs pour `runmqsc`», à la page 22.

## Pourquoi et quand exécuter cette tâche

La commande **runmqsc** permet d'ouvrir une invite de commande à partir de laquelle vous pouvez émettre des commandes MQSC. Ces commandes et leur syntaxe sont décrites dans la [référence des commandes MQSC](#).

Lorsque vous démarrez l'invite de commande **runmqsc** comme décrit dans cette tâche, vous définissez l'invite de sorte qu'elle s'exécute dans l'un des trois modes, en fonction des indicateurs définis dans la commande:

- *Mode de vérification*, dans lequel les commandes MQSC sont vérifiées sur un gestionnaire de files d'attente local, mais ne sont pas exécutées.
- *Mode direct*, dans lequel les commandes MQSC sont exécutées sur un gestionnaire de files d'attente local.
- *Mode indirect*, dans lequel les commandes MQSC sont exécutées sur un gestionnaire de files d'attente éloignées.

La procédure ci-dessous définit l'invite à exécuter en mode direct. D'autres options sont illustrées dans les exemples qui suivent les principales étapes.

## Procédure

1. Ouvrez une fenêtre de commande ou un shell et entrez la commande suivante:

```
runmqsc QMgrName
```

Où *QMgrName* indique le nom du gestionnaire de files d'attente devant traiter les commandes MQSC. Vous pouvez laisser la zone *QMgrName* vide pour traiter les commandes MQSC sur le gestionnaire de files d'attente par défaut.

2. Entrez les commandes MQSC requises. Par exemple, pour créer une file d'attente locale appelée ORANGE . LOCAL . QUEUE , entrez la commande suivante:

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

Pour les commandes comportant trop de paramètres pour tenir sur une seule ligne, utilisez des caractères de continuation pour indiquer qu'une commande se poursuit sur la ligne suivante:

- Le signe moins (-) indique que la commande doit être poursuivie à partir du début de la ligne suivante.
- Un signe plus (+) indique que la commande doit être poursuivie à partir du premier caractère non blanc sur la ligne suivante.

L'entrée de commande se termine par le caractère final d'une ligne non vide qui n'est pas un caractère de continuation. Vous pouvez également arrêter l'entrée de commande de manière explicite en entrant un point-virgule (;).

3. Arrêtez d'utiliser les commandes MQSC en entrant la commande suivante:

```
end
```

Vous pouvez également utiliser la commande **exit** , la commande **quit** ou le caractère EOF de votre système d'exploitation.

## Résultats

Lorsque vous émettez des commandes MQSC, le gestionnaire de files d'attente renvoie des messages d'opérateur qui confirment vos actions ou vous indiquent les erreurs que vous avez commises. Par exemple, le message suivant confirme la création d'une file d'attente:

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

Le message suivant indique que vous avez fait une erreur de syntaxe:

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

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

Ces messages sont envoyés à l'unité de sortie standard, qui est par défaut l'affichage. Si vous n'avez pas entré la commande correctement, reportez-vous aux informations de référence de la commande pour trouver la syntaxe correcte. Voir [MQSC commands reference](#).

### Exemple

Voici des variantes de la commande `runmqsc QMgrName` utilisée dans les étapes précédentes. Ces variantes créent différentes configurations de l'invite de commande **runmqsc**.

- La commande suivante utilise le filtrage de commande pour transmettre une seule commande MQSC à l'interpréteur MQSC.

Sous Windows :

```
echo display chstatus(*) | runmqsc QMname
```

Sous Linux :

```
echo "display chstatus(*)" | runmqsc QMname
```

- La commande suivante ne spécifiant pas de nom de gestionnaire de files d'attente, les commandes MQSC sont traitées sur le gestionnaire de files d'attente par défaut.

```
runmqsc
```

- Cette commande soumet des commandes au gestionnaire de files d'attente QMREMOTE, en utilisant QMLOCAL pour soumettre les commandes.

```
runmqsc -w 30 -m QMLOCAL QMREMOTE
```

- Cette commande vérifie que la syntaxe de commande est correcte sur un gestionnaire de files d'attente local sans exécuter les commandes. Notez que les commandes à vérifier sont lues à partir d'un fichier d'entrée `myprog.in`.

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

Pour plus d'informations sur l'utilisation des fichiers d'entrée et de sortie, voir [«Exécution de commandes MQSC à partir de fichiers texte sous runmqsc»](#), à la page 23.

## Que faire ensuite

Pour plus de détails sur la syntaxe de la commande **runmqsc** , les paramètres facultatifs et les codes retour, voir [runmqsc \(exécution de commandes MQSC\)](#).

### Tâches associées

«Exécution de commandes MQSC à partir de fichiers texte sous runmqsc», à la page 23

Si vous disposez de commandes longues ou que vous utilisez une séquence de commandes particulière à plusieurs reprises, vous pouvez utiliser un fichier texte pour émettre des commandes MQSC. Vous pouvez rediriger `stdin` à partir d'un fichier texte. Vous pouvez également rediriger la sortie vers un fichier.

### Référence associée

[Informations de référence sur les commandes MQSC](#)

## ALW Définition de l'invite de commande MQSC

Sous AIX, Linux, and Windows, utilisez la variable d'environnement **MQPROMPT** pour définir l'invite qui s'affiche lorsque vous exécutez la commande **runmqsc** . Il est ainsi plus facile de voir que vous êtes dans un environnement MQSC et d'afficher certains détails de l'environnement en cours.

## Pourquoi et quand exécuter cette tâche

Vous pouvez définir l'invite qui s'affiche lorsque vous exécutez la commande **runmqsc** . L'invite est insérée à la fois lorsque la commande **runmqsc** est exécutée en mode interactif et lorsque l'entrée est redirigée vers **runmqsc** à partir d'un fichier ou du périphérique d'entrée standard (`stdin`).

Vous pouvez inclure du texte en clair dans l'invite de commande et vous pouvez également insérer des variables d'environnement en utilisant la notation `+VARNAME+` de la même manière que les définitions d'objet de service IBM MQ . Pour plus d'informations, voir «[Utilisation d'insertions remplaçables sur des définitions de service](#)», à la page 196.

Il existe un certain nombre d'autres insertions remplaçables supplémentaires fournies par IBM MQ , décrites dans le tableau suivant.

Insertion remplaçable	Description
<b>MQ_HOST_NAME</b>	Nom d'hôte du système
<b>MQ_FILE_SEP</b>	Séparateur de fichiers spécifique à la plateforme: <ul style="list-style-type: none"><li>▶ Linux ▶ AIX Sur les systèmes AIX and Linux , <b>MQ_FILE_SEP</b> est / .</li><li>▶ Windows Sur les systèmes Windows , l'emplacement du <b>MQ_FILE_SEP</b> est \</li></ul>
<b>MQ_PATH_SEP</b>	Séparateur de chemin spécifique à la plateforme: <ul style="list-style-type: none"><li>▶ Linux ▶ AIX Sur les systèmes AIX and Linux , <b>MQ_PATH_SEP</b> est : .</li><li>▶ Windows Sur les systèmes Windows , l'emplacement du <b>MQ_PATH_SEP</b> est ;</li></ul>
<b>MQ_DATE_TIME</b>	Date et heure du système local dans un format YYYY-MM-DD hh:mm:ss . SSS fixe, par exemple:  2020-12-25 17:41:37.408

### Remarques :

- Les valeurs d'insertion MQ remplaçables sont liées à l'installation IBM MQ et au système hôte auquel la commande **runmqsc** est associée.

- **MQPROMPT** est limité à un maximum de 256 caractères lorsque les insertions sont développées. Les extensions **MQPROMPT** sur cette valeur entraînent la troncature de l'intégralité de la chaîne **MQPROMPT** sans les extensions.

## Exemple

L'exemple suivant définit l'invite sur MQSC:

-  

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

- 

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

- 

L'exemple suivant définit la variable **MQPROMPT** sur un système AIX . L'invite est définie pour afficher un nom d'utilisateur (extrait de la variable d'environnement système associée), le nom du gestionnaire de files d'attente et le nom d'hôte IBM MQ (extrait des insertions remplaçables IBM MQ) :

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

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

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

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

L'exemple suivant ajoute un horodatage aux exemples **MQPROMPT** ci-dessus, extrait des insertions remplaçables MQ :

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

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

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

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

- 

- 

## Activation du rappel et de l'exécution des commandes et des touches de commande Emacs pour runmqsc

Utilisez l'invite de commande **runmqsc** sous AIX and Linux pour activer les touches de commande de rappel de commande, d'exécution de commande et Emacs.



### Pourquoi et quand exécuter cette tâche

Sur les systèmes AIX and Linux , vous pouvez rendre disponibles les fonctions d'éditeur de ligne de commande supplémentaires suivantes à partir de l'invite de commande **runmqsc** :

- Rappel des commandes précédemment entrées à l'aide des touches de déplacement vers le haut et vers le bas

- Exécution automatique du mot clé suivant d'une commande à l'aide de la touche de tabulation et de la barre d'espace
- Touches de commande **Emacs** ou fonctions de touche de commande similaires

Pour utiliser ces fonctions, la bibliothèque `curses` doit être installée. Si la bibliothèque `curses` n'est pas installée sur votre système, l'invite `runmqsc` ne dispose pas des fonctions de l'éditeur de ligne de commande et un message s'affiche lorsque l'invite de commande `runmqsc` est démarrée. Le nom de la bibliothèque `curses` à installer dépend de la plateforme UNIX :

-  Sous AIX, installez `curses`.
-  Sous Linux, installez `ncurses`.

## Procédure

- Installez `ncurses` ou `curses`.

**Remarque :** L'exemple suivant utilise des instructions pour Linux

Exécutez la commande suivante pour rechercher les packages `ncurses` existants:

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

Les packages `ncurses` requis sont les suivants:

```
ncurses-term-6.1-7.20180224.e18.noarch
ncurses-6.1-7.20180224.e18.x86_64
ncurses-base-6.1-7.20180224.e18.noarch
ncurses-c++-libs-6.1-7.20180224.e18.x86_64
ncurses-libs-6.1-7.20180224.e18.x86_64
ncurses-compat-libs-6.1-7.20180224.e18.x86_64
ncurses-devel-6.1-7.20180224.e18.x86_64
```

Vous pouvez installer tous les packages `ncurses` requis répertoriés dans le texte précédent en exécutant la commande suivante:

```
yum install ncurses*
```

- Personnalisez les liaisons de touches Emacs.

Vous pouvez personnaliser les clés qui sont liées aux commandes. Par exemple, vous pouvez lier les touches à des liaisons vi au lieu des liaisons de touches Emacs par défaut.

Les clés sont personnalisées en éditant le fichier `.editrc` qui est stocké dans le répertoire de base. Pour plus d'informations, voir `editrc` dans les pages d'aide FreeBSD .

- Désactiver le rappel de commande, l'exécution de commande et les touches de commande Emacs.

Pour ce faire, définissez la variable d'environnement **MQ\_OVERRIDE\_LIBEDIT\_LOAD** sur TRUE.

Cette variable d'environnement peut être utilisée comme solution palliative lorsque l'invite de commande `runmqsc` affiche le message d'information suivant:

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

## Exécution de commandes MQSC à partir de fichiers texte sous `runmqsc`

Si vous disposez de commandes longues ou que vous utilisez une séquence de commandes particulière à plusieurs reprises, vous pouvez utiliser un fichier texte pour émettre des commandes MQSC. Vous pouvez rediriger `stdin` à partir d'un fichier texte. Vous pouvez également rediriger la sortie vers un fichier.

## Avant de commencer

Cette tâche suppose que vous avez créé un fichier texte contenant les commandes MQSC que vous souhaitez exécuter. Pour obtenir une syntaxe détaillée et des exemples de ces fichiers, voir [«Syntaxe du fichier d'entrée MQSC»](#), à la page 16.

Vous pouvez définir l'invite de commande MQSC sur une invite de votre choix à l'aide de la variable d'environnement **MQPROMPT**. Pour plus d'informations, voir [«Définition de l'invite de commande MQSC»](#), à la page 21.

## Pourquoi et quand exécuter cette tâche

L'entrée de la commande **runmqsc** est extraite de l'*unité d'entrée standard*, également appelée `stdin`. Il s'agit généralement du clavier, mais vous pouvez spécifier que l'entrée doit provenir d'un port série ou d'un fichier.

La sortie de la commande **runmqsc** est envoyée à l'*unité de sortie standard*, également appelée `stdout`. Il s'agit généralement d'un affichage, mais vous pouvez rediriger la sortie vers un port série ou un fichier.

## Procédure

1. Sur un gestionnaire de files d'attente local, vérifiez que la syntaxe de commande dans le fichier est correcte sans exécuter les commandes.

Utilisez l'indicateur **-v** dans la commande **runmqsc**, avec l'une des options suivantes:

- Utilisez l'option **-f** pour identifier le nom de fichier du texte d'entrée. Exemple :

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

Vous ne pouvez pas spécifier de gestionnaire de files d'attente éloignées lors de la vérification des commandes. C'est-à-dire que vous ne pouvez pas spécifier l'indicateur **-w**.

Le rapport renvoyé est similaire à celui présenté dans la [Figure 2](#), à la page 18.

2. Lorsque la syntaxe de la commande est correcte, supprimez l'indicateur **-v**, puis réexécutez la commande **runmqsc**.

Notez que vous pouvez désormais spécifier un gestionnaire de files d'attente éloignées.

- Exécutez (par exemple) la commande suivante:

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

Figure 1, à la page 17 montre une extraction à partir d'un fichier de commandes tel que `myprog.in` et Figure 2, à la page 18 montre l'extraction correspondante de la sortie à partir d'un fichier de rapport tel que `results.out`.

## Que faire ensuite

Pour plus de détails sur la syntaxe de la commande **runmqsc**, les paramètres facultatifs et les codes retour, voir [runmqsc \(exécution de commandes MQSC\)](#).

### Tâches associées

[«Définition de l'invite de commande MQSC»](#), à la page 21

Sous AIX, Linux, and Windows, utilisez la variable d'environnement **MQPROMPT** pour définir l'invite qui s'affiche lorsque vous exécutez la commande **runmqsc**. Il est ainsi plus facile de voir que vous êtes dans un environnement MQSC et d'afficher certains détails de l'environnement en cours.

[«Exécution interactive de commandes MQSC sous runmqsc»](#), à la page 18

Sous AIX, Linux, and Windows, vous pouvez utiliser l'invite de commande **runmqsc** pour émettre des commandes MQSC à un gestionnaire de files d'attente de manière interactive. L'exécution interactive est particulièrement adaptée aux tests rapides.



## Référence associée

[Informations de référence sur les commandes MQSC](#)

Multi

## Configuration automatique à partir d'un script MQSC au démarrage

Vous pouvez configurer votre gestionnaire de files d'attente pour qu'il applique automatiquement le contenu d'un script MQSC ou d'un ensemble de scripts MQSC à chaque démarrage du gestionnaire de files d'attente.

Vous pouvez utiliser cette fonctionnalité pour avoir une configuration qui peut être modifiée et réexécutée automatiquement au prochain redémarrage du gestionnaire de files d'attente. Par exemple, si le ou les scripts se trouvent sur une unité montée, il est possible de disposer d'une configuration centralisée dans laquelle la dernière version est appliquée à chaque gestionnaire de files d'attente au fur et à mesure de son démarrage.

Un scénario particulier dans lequel cela peut être utile consiste à s'assurer qu'un cluster uniforme contient les mêmes définitions sur tous les gestionnaires de files d'attente du cluster, en disposant d'un ensemble unique de configurations qu'ils appliquent tous. Pour un exemple, voir [Création d'un cluster uniforme](#).

### Avant de commencer

Vous pouvez utiliser :

1. Un script unique et créez un fichier texte à l'aide des commandes MQSC.
2. Un ensemble de scripts MQSC:
  - Pour identifier un répertoire dans lequel les configurations existeront, et
  - Dans ce répertoire, créez des fichiers, chacun avec l'extension `.mqsc`, par exemple `queues.mqsc`.

Etant donné que ce script est réappliqué à chaque démarrage du gestionnaire de files d'attente, il est important que les commandes puissent être réexécutées. Par exemple, une commande **DEFINE** doit inclure la chaîne **REPLACE**, sinon la commande apparaît comme un échec au deuxième démarrage du gestionnaire de files d'attente, car l'objet existe déjà.

Notez que dans un script MQSC, toute ligne préfixée avec `*` est traitée comme un commentaire.

### Activation de la configuration automatique des scripts MQSC

**Important :** Vous ne devez pas émettre de commandes pour les canaux de type MQTT, car ils ne sont pas pris en charge pour la configuration automatique lors du démarrage.

Vous pouvez configurer un nouveau gestionnaire de files d'attente en utilisant l'indicateur **-ic** dans la commande `crtnmqm` et en pointant vers un fichier ou un répertoire spécifique. La valeur fournie est stockée dans le fichier `qm.ini` sous la section `AutoConfig`, en tant qu'attribut **MQSCConfig**.

Vous pouvez configurer un gestionnaire de files d'attente existant pour activer la configuration MQSC automatique en ajoutant l'attribut de strophe `AutoConfig MQSCConfig`, qui pointe vers un fichier ou un répertoire valide. Exemple :

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

### Comment la configuration automatique fonctionne-t-elle?

Lors du démarrage du gestionnaire de files d'attente, la configuration identifiée par l'attribut de section **MQSCConfig** est transmise via la validation `runmqsc`, afin de garantir une syntaxe valide, puis stockée dans l'arborescence de données du gestionnaire de files d'attente dans le répertoire `autocfg` en tant que fichier unique `cached.mqsc`.

Lorsque plusieurs fichiers d'un répertoire sont traités, ils sont traités par ordre alphabétique et s'ils contiennent une commande MQSC end ou quit, le reste du contenu de ce fichier est ignoré.

Lors du premier démarrage du gestionnaire de files d'attente, une impossibilité de lire le fichier ou le répertoire, ou un fichier dont la syntaxe MQSC n'est pas valide, empêche le démarrage du gestionnaire de files d'attente, avec un message d'erreur approprié à la fois sur la console et dans le journal des erreurs du gestionnaire de files d'attente.

Lors des redémarrages suivants, si le fichier ou le répertoire désigné est illisible ou contient une syntaxe MQSC non valide, le fichier précédemment mis en cache est utilisé et un message écrit dans le journal des erreurs du gestionnaire de files d'attente le met en évidence.

Au moment où le contenu de `cached.mqsc` est appliqué au gestionnaire de files d'attente, lorsque toutes les commandes MQSC ont été appliquées, le gestionnaire de files d'attente est activé pour que les applications se connectent. Le journal `runmqsc` de la configuration appliquée est stocké dans le répertoire des erreurs du gestionnaire de files d'attente, sous la forme d'un fichier appelé `autocfgmqsc.LOG`.

En outre, toute commande MQSC qui ne s'exécute pas correctement est consignée dans le journal des erreurs du gestionnaire de files d'attente, en identifiant la raison de l'échec de la commande.

## Automatisation de l'administration de IBM MQ à l'aide des commandes PCF

---

Vous pouvez décider qu'il serait avantageux pour votre installation d'automatiser certaines tâches d'administration et de surveillance. Vous pouvez automatiser les tâches d'administration pour les gestionnaires de files d'attente locales et éloignées à l'aide des commandes PCF (Programmable Command Format). Cette section suppose que vous avez l'expérience de l'administration des objets IBM MQ.

### Commandes PCF

Les commandes PCF (IBM MQ programmable command format) peuvent être utilisées pour programmer des tâches d'administration dans un programme d'administration. Ainsi, à partir d'un programme, vous pouvez manipuler des objets de gestionnaire de files d'attente (files d'attente, définitions de processus, listes de noms, canaux, canaux de connexion client, programmes d'écoute, services et objets d'informations d'authentification) et même manipuler les gestionnaires de files d'attente eux-mêmes.

Les commandes PCF couvrent la même gamme de fonctions que les commandes MQSC. Vous pouvez écrire un programme pour émettre des commandes PCF vers n'importe quel gestionnaire de files d'attente du réseau à partir d'un noeud unique. De cette manière, vous pouvez à la fois centraliser et automatiser les tâches d'administration.

Chaque commande PCF est une structure de données imbriquée dans la partie données d'application d'un message IBM MQ. Chaque commande est envoyée au gestionnaire de files d'attente cible à l'aide de la fonction MQI MQPUT de la même manière que tout autre message. A condition que le serveur de commandes soit en cours d'exécution sur le gestionnaire de files d'attente qui reçoit le message, le serveur de commandes l'interprète comme un message de commande et exécute la commande. Pour obtenir les réponses, l'application émet un appel MQGET et les données de réponse sont renvoyées dans une autre structure de données. L'application peut ensuite traiter la réponse et agir en conséquence.

**Remarque :** Contrairement aux commandes MQSC, les commandes PCF et leurs réponses ne sont pas au format texte que vous pouvez lire.

En bref, voici quelques-uns des éléments nécessaires à la création d'un message de commande PCF:

#### Descripteur de message

Il s'agit d'un descripteur de message IBM MQ standard, dans lequel:

- Le type de message (*MsgType*) est MQMT\_REQUEST.
- Le format de message (*Format*) est MQFMT\_ADMIN.

## Données d'application

Contient le message PCF incluant l'en-tête PCF, dans lequel:

- Le type de message PCF (*Type*) indique MQCFT\_COMMAND.
- L'identificateur de la commande indique la commande, par exemple *Change Queue* (MQCMD\_CHANGE\_Q).

Pour obtenir une description complète des structures de données PCF et savoir comment les implémenter, voir [«Introduction à IBM MQ Programmable Command Formats»](#), à la page 27.

## Attributs d'objet PCF

Les attributs d'objet dans PCF ne sont pas limités à huit caractères, comme pour les commandes MQSC. Ils sont présentés dans ce guide en italique. Par exemple, l'équivalent PCF de RQMNAME est *RemoteQMgrName*.

## Echapper les fichiers PCF

Les fichiers PCF d'arrêt programme sont des commandes PCF qui contiennent des commandes MQSC dans le texte du message. Vous pouvez utiliser des fichiers PCF pour envoyer des commandes à un gestionnaire de files d'attente éloignées. Pour plus d'informations sur les fichiers PCF d'échappement, voir [Echap](#).

## Introduction à IBM MQ Programmable Command Formats

Les formats PCF (Programmable Command Formats) définissent les messages de commande et de réponse qui peuvent être échangés entre un programme et n'importe quel gestionnaire de files d'attente (qui prend en charge les PCF) d'un réseau. Les fichiers PCF simplifient l'administration du gestionnaire de files d'attente et d'autres fonctions d'administration du réseau. Ils peuvent être utilisés pour résoudre le problème de l'administration complexe des réseaux distribués, en particulier à mesure que les réseaux augmentent en taille et en complexité.

Les formats de commande programmables sont pris en charge sur toutes les plateformes IBM MQ .

## Résolution des incidents liés aux commandes PCF

L'administration des réseaux distribués peut devenir complexe. Les problèmes d'administration continuent de croître à mesure que les réseaux augmentent en taille et en complexité.

Exemples d'administration spécifique à la messagerie et à la mise en file d'attente:

- Gestion des ressources.

Par exemple, la création et la suppression de files d'attente.

- Surveillance des performances.

Par exemple, la longueur maximale de la file d'attente ou le débit des messages.

- Contrôle.

Par exemple, l'optimisation des paramètres de file d'attente tels que la longueur maximale de la file d'attente, la longueur maximale des messages et l'activation et la désactivation des files d'attente.

- Routage des messages.

Définition de routes alternatives à travers un réseau.

Les commandes PCF IBM MQ peuvent être utilisées pour simplifier l'administration du gestionnaire de files d'attente et d'autres fonctions d'administration du réseau. Les commandes PCF vous permettent d'utiliser une application unique pour effectuer l'administration du réseau à partir d'un seul gestionnaire de files d'attente au sein du réseau.

## Que sont les PCF?

Les PCF définissent les messages de commande et de réponse qui peuvent être échangés entre un programme et n'importe quel gestionnaire de files d'attente (qui prend en charge les PCF) d'un réseau. Vous pouvez utiliser des commandes PCF dans un programme d'application de gestion des systèmes pour l'administration des objets IBM MQ : objets d'informations d'authentification, canaux, programmes d'écoute de canal, listes de noms, définitions de processus, gestionnaires de files d'attente, files d'attente, services et classes de stockage. L'application peut fonctionner à partir d'un point unique du réseau pour communiquer des informations de commande et de réponse avec n'importe quel gestionnaire de files d'attente, local ou distant, à l'aide du gestionnaire de files d'attente local.


Chaque gestionnaire de files d'attente possède une file d'attente d'administration avec un nom de file d'attente standard et votre application peut envoyer des messages de commande PCF à cette file d'attente. Chaque gestionnaire de files d'attente dispose également d'un serveur de commandes pour traiter les messages de commande de la file d'attente d'administration. Les messages de commande PCF peuvent donc être traités par n'importe quel gestionnaire de files d'attente du réseau et les données de réponse peuvent être renvoyées à votre application à l'aide de la file d'attente de réponses spécifiée. Les commandes PCF et les messages de réponse sont envoyés et reçus à l'aide de l'interface MQI (Message Queue Interface) normale.

Pour obtenir la liste des commandes PCF disponibles, y compris leurs paramètres, voir [Définitions des formats de commande programmables](#).

## Utilisation des formats de commande programmables IBM MQ

Vous pouvez utiliser des fichiers PCF dans un programme de gestion des systèmes pour l'administration à distance de IBM MQ .

Cette section contient :


- [«Messages de commande PCF»](#), à la page 28
- [«Réponses PCF dans IBM MQ»](#), à la page 31
-  [«Extended responses»](#), à la page 33
- [Règles d'appellation des objets IBM MQ](#)
- [«Vérification des droits d'accès pour les commandes PCF dans IBM MQ»](#), à la page 34


### Messages de commande PCF

Les messages de commande PCF sont constitués d'un en-tête PCF, de paramètres identifiés dans cet en-tête et de données de message définies par l'utilisateur. Les messages sont émis à l'aide d'appels d'interface de file d'attente de messages.

Chaque commande et ses paramètres sont envoyés sous la forme d'un message de commande distinct contenant un en-tête PCF suivi d'un certain nombre de structures de paramètres ; pour plus de détails sur l'en-tête PCF, voir [MQCFH-PCF header](#) pour un exemple de structure de paramètre, voir [MQCFST-PCF string parameter](#). L'en-tête PCF identifie la commande et le nombre de structures de paramètres qui suivent dans le même message. Chaque structure de paramètres fournit un paramètre à la commande.

Les réponses aux commandes, générées par le serveur de commandes, ont une structure similaire. Il existe un en-tête PCF, suivi d'un certain nombre de structures de paramètres. Les réponses peuvent consister en plusieurs messages, mais les commandes consistent toujours en un seul message.

 Sous Multiplateformes, la file d'attente à laquelle les commandes PCF sont envoyées est toujours appelée SYSTEM.ADMIN.COMMAND.QUEUE.

 Sous z/OS, les commandes sont envoyées à SYSTEM.COMMAND.INPUT, bien que SYSTEM.ADMIN.COMMAND.QUEUE peut être un alias pour lui. Le serveur de commandes gérant cette file d'attente envoie les réponses à la file d'attente définie par les zones *ReplyToQ* et *ReplyToQMGr* dans le descripteur de message du message de commande.

## Comment émettre des messages de commande PCF

Utilisez les appels MQI (Message Queue Interface) normaux, MQPUT, MQGET, etc., pour insérer et extraire des messages de commande et de réponse PCF vers et depuis leurs files d'attente.

### Remarque :

Vérifiez que le serveur de commandes est en cours d'exécution sur le gestionnaire de files d'attente cible pour que la commande PCF puisse être traitée sur ce gestionnaire de files d'attente.

Pour obtenir la liste des fichiers d'en-tête fournis, voir [IBM MQ COPY, header, include et module files](#).

## Descripteur de message pour une commande PCF

Le descripteur de message IBM MQ est intégralement documenté dans [MQMD-Descripteur de message](#).

Un message de commande PCF contient les zones suivantes dans le descripteur de message:

### Rapport

Toute valeur valide, selon les besoins.

### MsgType

Cette zone doit être MQMT\_REQUEST pour indiquer un message nécessitant une réponse.

### Expiration

Toute valeur valide, selon les besoins.

### Commentaires

Défini sur MQFB\_NONE

### Multi Codage

Si vous envoyez des données à un système IBM MQ for Multiplatforms , définissez cette zone sur le codage utilisé pour les données de message. La conversion est effectuée si nécessaire.

### Multi CodedCharSetId

Si vous envoyez des données à un système IBM MQ for Multiplatforms , définissez cette zone sur l'identificateur de jeu de caractères codés utilisé pour les données de message. La conversion est effectuée si nécessaire.

### Format

Défini sur MQFMT\_ADMIN.

### Priorité

Toute valeur valide, selon les besoins.

### Persistance

Toute valeur valide, selon les besoins.

### MsgId

L'application émettrice peut spécifier n'importe quelle valeur ou MQMI\_NONE peut être spécifié pour demander au gestionnaire de files d'attente de générer un identificateur de message unique.

### CorrelId

L'application émettrice peut spécifier n'importe quelle valeur ou MQCI\_NONE peut être spécifié pour indiquer qu'il n'y a pas d'identificateur de corrélation.

### ReplyToQ

Nom de la file d'attente de réception de la réponse.

### ReplyToQMgr

Nom du gestionnaire de files d'attente pour la réponse (ou vide).

### Zones de contexte de message

Ces zones peuvent être définies sur n'importe quelle valeur valide, selon les besoins. Normalement, l'option d'insertion de message MQPMO\_DEFAULT\_CONTEXT est utilisée pour définir les zones de contexte de message sur les valeurs par défaut.

Si vous utilisez une structure MQMD version-2 , vous devez définir les zones supplémentaires suivantes:

**GroupId**

Défini sur MQGI\_NONE

**MsgSeqNumber**

Définir sur 1

**Décalage**

Définir sur 0

**MsgFlags**

Défini sur MQMF\_NONE

**OriginalLength**

Défini sur MQOL\_UNDEFINED

**Envoi de données utilisateur**

Les structures PCF peuvent également être utilisées pour envoyer des données de message définies par l'utilisateur. Dans ce cas, la zone *Format* du descripteur de message doit être définie sur MQFMT\_PCF.

**Envoi et réception de messages PCF dans une file d'attente spécifiée****Envoi de messages PCF à une file d'attente spécifiée**

Pour envoyer un message à une file d'attente spécifiée, l'appel de sac mqPut convertit le contenu du sac spécifié en message PCF et envoie le message à la file d'attente spécifiée. Le contenu du sac est inchangé après l'appel.

Comme entrée pour cet appel, vous devez fournir:

- Un descripteur de connexion MQI.
- Descripteur d'objet de la file d'attente dans laquelle le message doit être placé.
- Descripteur de message. Pour plus d'informations sur le descripteur de message, voir [MQMD-Descripteur de message](#).
- Options d'insertion de message à l'aide de la structure MQPMO. Pour plus d'informations sur la structure MQPMO, voir [MQPMO-Options d'insertion de message](#).
- Poignée du sac à convertir en message.

**Remarque :** Si le sac contient un message d'administration et que l'appel d'interrogation mqAdda a été utilisé pour insérer des valeurs dans le sac, la valeur de l'élément de données MQIASY\_COMMAND doit être une commande INQUIRE reconnue par MQAI.

Pour une description complète de l'appel de sac mqPut, voir [mqPutBag](#).

**Réception de messages PCF d'une file d'attente spécifiée**

Pour recevoir un message d'une file d'attente spécifiée, l'appel mqGetBag extrait un message PCF d'une file d'attente spécifiée et convertit les données de message en un sac de données.

Comme entrée pour cet appel, vous devez fournir:

- Un descripteur de connexion MQI.
- Descripteur d'objet de la file d'attente à partir de laquelle le message doit être lu.
- Descripteur de message. Dans la structure MQMD, le paramètre **Format** doit être MQFMT\_ADMIN, MQFMT\_EVENT ou MQFMT\_PCF.

**Remarque :** Si le message est reçu dans une unité de travail (c'est-à-dire avec l'option MQGMO\_SYNCPOINT) et que le format du message n'est pas pris en charge, l'unité de travail peut être annulée. Le message est ensuite réintégré dans la file d'attente et peut être extrait à l'aide de

l'appel MQGET au lieu de l'appel Bag mqGet. Pour plus d'informations sur le descripteur de message, voir [Options MQGMO-Get-message](#).

- Obtenir les options de message à l'aide de la structure MQGMO. Pour plus d'informations sur la structure MQGMO, voir [MQMD-Descripteur de message](#).
- Poignée du sac devant contenir le message converti.

Pour une description complète de l'appel de sac mqGet, voir [mqGetBag](#).

## **Réponses PCF dans IBM MQ**

En réponse à chaque commande, le serveur de commandes génère un ou plusieurs messages de réponse. Un message de réponse a un format similaire à celui d'un message de commande.

L'en-tête PCF possède la même valeur d'identificateur de commande que la commande à laquelle il correspond à une réponse (voir [MQCFH-en-tête PCF](#) pour plus de détails). L'identificateur de message et l'identificateur de corrélation sont définis en fonction des options de rapport de la demande.

Si le type d'en-tête PCF du message de commande est MQCFT\_COMMAND, seules les réponses standard sont générées. Ces commandes sont prises en charge sur Multiplatforms uniquement. Les anciennes applications ne prennent pas en charge PCF sous z/OS ; IBM MQ Windows Explorer est l'une de ces applications (toutefois, l'explorateur IBM WebSphere MQ 6.0 ou ultérieur IBM MQ prend en charge PCF sous z/OS).

Si le type d'en-tête PCF du message de commande est MQCFT\_COMMAND\_XR, des réponses étendues ou standard sont générées. Ces commandes sont prises en charge sur z/OS et sur Multiplatforms. Les commandes émises sur z/OS génèrent uniquement des réponses étendues.

Si une seule commande spécifie un nom d'objet générique, une réponse distincte est renvoyée dans son propre message pour chaque objet correspondant. Pour la génération de réponse, une seule commande avec un nom générique est traitée comme plusieurs commandes individuelles (à l'exception de la zone de contrôle MQCFC\_LAST ou MQCFC\_NOT\_LAST). Sinon, un message de commande génère un message de réponse.

Certaines réponses PCF peuvent renvoyer une structure même si elle n'est pas demandée. Cette structure est illustrée dans la définition de la réponse ( [Définitions des formats de commande programmables](#) ) comme *toujours renvoyé*. La raison pour laquelle, pour ces réponses, il est nécessaire de nommer les objets dans la réponse pour identifier l'objet auquel les données s'appliquent.

## **Descripteur de message pour une réponse**

Un message de réponse comporte les zones suivantes dans le descripteur de message:

### **MsgType**

Cette zone est MQMT\_REPLY.

### **MsgId**

Cette zone est générée par le gestionnaire de files d'attente.

### **CorrelId**

Cette zone est générée en fonction des options de rapport du message de commande.

### **Format**

Cette zone est MQFMT\_ADMIN.

### **Codage**

Défini sur MQENC\_NATIVE.

### **CodedCharSetId**

Défini sur MQCCSI\_Q\_MGR.

### **Persistance**

Identique à celui du message de commande.

### **Priorité**

Identique à celui du message de commande.

La réponse est générée avec MQPMO\_PASS\_IDENTITY\_CONTEXT.

Les messages de commande avec un type d'en-tête MQCFT\_COMMAND, des réponses standard sont générées. Ces commandes sont prises en charge uniquement sur Multiplatforms.

Il existe trois types de réponse standard:

- Réponse OK
- Réponse d'erreur
- Réponse aux données

## Réponse OK

Cette réponse se compose d'un message commençant par un en-tête de format de commande, avec une zone *CompCode* de MQCC\_OK ou MQCC\_WARNING.

Pour MQCC\_OK, *Reason* est MQRC\_NONE.

Pour MQCC\_WARNING, *Reason* identifie la nature de l'avertissement. Dans ce cas, l'en-tête de format de commande peut être suivi d'une ou de plusieurs structures de paramètre d'avertissement appropriées à ce code anomalie.

Dans les deux cas, pour une commande d'interrogation, d'autres structures de paramètres peuvent suivre, comme décrit dans les sections suivantes.

## Réponse d'erreur

Si la commande comporte une erreur, un ou plusieurs messages de réponse d'erreur sont envoyés (plusieurs peuvent être envoyés même pour une commande qui n'aurait normalement qu'un seul message de réponse). Ces messages de réponse d'erreur ont MQCFC\_LAST ou MQCFC\_NOT\_LAST définis comme approprié.

Chaque message de ce type commence par un en-tête de format de réponse, avec une valeur *CompCode* de MQCC\_FAILED et une zone *Reason* qui identifie l'erreur particulière. En général, chaque message décrit une erreur différente. En outre, chaque message a soit zéro, soit une (jamais plus d'une) structure de paramètres d'erreur après l'en-tête. Cette structure de paramètre, s'il en existe une, est une structure MQCFIN, avec une zone *Parameter* contenant l'un des éléments suivants:

- ID\_PARAMÈTRE\_MQIACF\_ID

La zone *Value* de la structure correspond à l'identificateur du paramètre erroné (par exemple, MQCA\_Q\_NAME).

- ID\_ERREUR\_MQIACF

Cette valeur est utilisée avec une valeur *Reason* (dans l'en-tête de format de commande) de MQRC\_UNEXPECTED\_ERROR. La zone *Value* de la structure MQCFIN correspond au code anomalie inattendu reçu par le serveur de commandes.

- MQIACF\_SELECTOR

Cette valeur se produit si une structure de liste (MQCFIL) envoyée avec la commande contient un sélecteur en double ou non valide. La zone *Reason* dans l'en-tête de format de commande identifie l'erreur et la zone *Value* dans la structure MQCFIN est la valeur de paramètre dans la structure MQCFIL de la commande qui était erronée.

- MQIACF\_DÉCALAGE\_ERREUR\_ERREUR

Cette valeur se produit en cas d'erreur de comparaison de données dans la commande Ping Channel. La zone *Value* de la structure correspond au décalage de l'erreur de comparaison du canal ping.

- MQIA\_CODED\_CHAR\_SET\_ID

Cette valeur se produit lorsque l'identificateur de jeu de caractères codés dans le descripteur de message du message de commande PCF entrant ne correspond pas à celui du gestionnaire de files



d'attente cible. La zone *Value* de la structure correspond à l'identificateur de jeu de caractères codés du gestionnaire de files d'attente.

Le dernier (ou seul) message de réponse d'erreur est une réponse récapitulative, avec une zone *CompCode* de MQCC\_FAILED et une zone *Reason* de MQRCCF\_COMMAND\_FAILED. Ce message n'a pas de structure de paramètre à la suite de l'en-tête.

## Réponse aux données

Cette réponse consiste en une réponse OK (comme décrit précédemment) à une commande d'interrogation. La réponse OK est suivie de structures supplémentaires contenant les données demandées, comme décrit dans la rubrique [Définitions des formats de commande programmables](#).

Les applications ne doivent pas dépendre du renvoi de ces structures de paramètres supplémentaires dans un ordre particulier.

### *Extended responses*

Commands issued on z/OS generate extended responses.

There are three types of extended response:

- Message response, with type MQCFT\_XR\_MSG
- Item response, with type MQCFT\_XR\_ITEM
- Summary response, with type MQCFT\_XR\_SUMMARY

Each command can generate one, or more, sets of responses. Each set of responses comprises one or more messages, numbered sequentially from 1 in the *MsgSeqNumber* field of the PCF header. The *Control* field of the last (or only) response in each set has the value MQCFC\_LAST. For all other responses in the set, this value is MQCFC\_NOT\_LAST.

Any response can include one, or more, optional MQCFBS structures in which the *Parameter* field is set to MQBACF\_RESPONSE\_SET, the value being a response set identifier. Identifiers are unique and identify the set of responses which contain the response. For every set of responses, there is an MQCFBS structure that identifies it.

Extended responses have at least two parameter structures:

- An MQCFBS structure with the *Parameter* field set to MQBACF\_RESPONSE\_ID. The value in this field is the identifier of the set of responses to which the response belongs. The identifier in the first set is arbitrary. In subsequent sets, the identifier is one previously notified in an MQBACF\_RESPONSE\_SET structure.
- An MQCFST structure with the *Parameter* field set to MQCACF\_RESPONSE\_Q\_MGR\_NAME, the value being the name of the queue manager from which the set of responses come.

Many responses have additional parameter structures, and these structures are described in the following sections.

You cannot determine in advance how many responses there are in a set other than by getting responses until one with MQCFC\_LAST is found. Neither can you determine in advance how many sets of responses there are as any set might include MQBACF\_RESPONSE\_SET structures to indicate that additional sets are generated.

## Extended responses to Inquire commands

Inquire commands normally generate an item response (type MQCFT\_XR\_ITEM) for each item found that matches the specified search criteria. The item response has a *CompCode* field in the header with a value of MQCC\_OK, and a *Reason* field with a value of MQRC\_NONE. It also includes other parameter structures describing the item and its requested attributes, as described in [Definitions of the Programmable Command Formats](#).

If an item is in error, the *CompCode* field in the header has a value of MQCC\_FAILED and the *Reason* field identifies the particular error. Additional parameter structures are included to identify the item.

Certain Inquire commands might return general (not name-specific) message responses in addition to the item responses. These responses are informational, or error, responses of the type MQCFT\_XR\_MSG.

If the Inquire command succeeds, there might, optionally, be a summary response (type MQCFT\_XR\_SUMMARY), with a *CompCode* value of MQCC\_OK, and a *Reason* field value of MQRC\_NONE.

If the Inquire command fails, item responses might be returned, and there might optionally be a summary response (type MQCFT\_XR\_SUMMARY), with a *CompCode* value of MQCC\_FAILED, and a *Reason* field value of MQRCCF\_COMMAND\_FAILED.

## Extended responses to commands other than Inquire

Successful commands generate message responses in which the *CompCode* field in the header has a value of MQCC\_OK, and the *Reason* field has a value of MQRC\_NONE. There is always at least one message; it might be informational (MQCFT\_XR\_MSG) or a summary (MQCFT\_XR\_SUMMARY). There might optionally be additional informational (type MQCFT\_XR\_MSG) messages. Each informational message might include a number of additional parameter structures with information about the command; see the individual command descriptions for the structures that can occur.

Commands that fail generate error message responses (type MQCFT\_XR\_MSG), in which the *CompCode* field in the header has a value of MQCC\_FAILED and the *Reason* field identifies the particular error. Each message might include a number of additional parameter structures with information about the error: see the individual error descriptions for the structures that can occur. Informational message responses might be generated. There might, optionally, be a summary response (MQCFT\_XR\_SUMMARY), with a *CompCode* value of MQCC\_FAILED, and a *Reason* field value of MQRCCF\_COMMAND\_FAILED.

## Extended responses to commands using CommandScope

If a command uses the **CommandScope** parameter, or causes a command using the **CommandScope** parameter to be generated, there is an initial response set from the queue manager where the command was received. Then a separate set, or sets, of responses is generated for each queue manager to which the command is directed (as if multiple individual commands were issued). Finally, there is a response set from the receiving queue manager which includes an overall summary response (type MQCFT\_XR\_SUMMARY). The MQCACF\_RESPONSE\_Q\_MGR\_NAME parameter structure identifies the queue manager that generates each set.


The initial response set has the following additional parameter structures:

- MQIACF\_COMMAND\_INFO (MQCFIN). Possible values in this structure are MQCMDI\_CMDScope\_ACCEPTED or MQCMDI\_CMDScope\_GENERATED.
- MQIACF\_CMDScope\_Q\_MGR\_COUNT (MQCFIN). This structure indicates the number of queue managers to which the command is sent.

## Vérification des droits d'accès pour les commandes PCF dans IBM MQ

Lorsqu'une commande PCF est traitée, le *UserIdentifiant* du descripteur de message dans le message de commande est utilisé pour les vérifications des droits d'accès aux objets IBM MQ requises. La vérification des droits d'accès est implémentée différemment sur chaque plateforme, comme décrit dans cette rubrique.

Les vérifications sont effectuées sur le système sur lequel la commande est en cours de traitement ; par conséquent, cet ID utilisateur doit exister sur le système cible et disposer des droits requis pour traiter la commande. Si le message provient d'un système distant, l'un des moyens d'obtenir l'ID existant sur le système cible consiste à disposer d'un ID utilisateur correspondant sur les systèmes local et distant.

**Remarque :**  Pour plus d'informations sur le contrôle des droits d'accès sur z/OS, voir [Tâche 1: Identifier les paramètres système z/OS](#).

## IBM MQ for IBM i



Pour pouvoir traiter une commande PCF, l'ID utilisateur doit disposer des droits *dsp* sur l'objet IBM MQ sur le système cible.

En outre, des vérifications des droits d'accès aux objets IBM MQ sont effectuées pour certaines commandes PCF, comme illustré dans la [Tableau 2](#), à la page 36.

Dans la plupart des cas, ces vérifications sont les mêmes que celles effectuées par les commandes CL IBM MQ équivalentes émises sur un système local. Pour plus d'informations sur le mappage des droits IBM MQ vers les droits système IBM i et sur les droits requis pour les commandes CL IBM MQ, voir [Configuration de la sécurité sur IBM i](#). Les détails de la sécurité concernant les exits sont fournis dans la documentation [Link level security using a security exit](#).

**Pour traiter l'une des commandes suivantes**, l'ID utilisateur doit être membre du profil de groupe QMQMADM:

- Envoyer une commande PING à un canal
- Modifier un canal
- Copier un canal
- Créer un canal
- Supprimer un canal
- Réinitialisation du canal
- Résolution du canal
- Démarrer un canal
- Arrêter le canal
- Démarrer un initialisateur de canal
- Démarrer un programme d'écoute de canaux

## IBM MQ for UNIX, Linux, and Windows



Pour traiter une commande PCF, l'ID utilisateur doit disposer des droits *dsp* sur l'objet gestionnaire de files d'attente sur le système cible. En outre, des vérifications des droits d'accès aux objets IBM MQ sont effectuées pour certaines commandes PCF, comme illustré dans la [Tableau 2](#), à la page 36.

**Pour traiter l'une des commandes suivantes**, l'ID utilisateur doit appartenir au groupe *mqm*.

**Remarque :** Pour Windows **uniquement**, l'ID utilisateur peut appartenir au groupe *Administrators* ou au groupe *mqm*.

- Modifier un canal
- Copier un canal
- Créer un canal
- Supprimer un canal
- Envoyer une commande PING à un canal
- Réinitialisation du canal
- Démarrer un canal
- Arrêter le canal
- Démarrer un initialisateur de canal
- Démarrer un programme d'écoute de canaux
- Résolution du canal
- Réinitialisation d'un cluster
- Régénérer un cluster
- Interrompre un gestionnaire de files d'attente

- Reprendre un gestionnaire de files d'attente

## Droits d'accès aux objets IBM MQ pour Multiplatforms



Tableau 2. Droits sur les objets

Commande	IBM MQ Les droits d'accès d'un objet	Droits sur les classes (pour le type d'objet)
Modifier des informations d'authentification	dsp et chg	Non applicable
Modifier un canal	dsp et chg	Non applicable
Modifier le programme d'écoute de canal	dsp et chg	Non applicable
Modifier le canal de connexion client	dsp et chg	Non applicable
Modifier une liste de noms	dsp et chg	Non applicable
Changement de processus	dsp et chg	Non applicable
Modifier une file d'attente	dsp et chg	Non applicable
Modifier un gestionnaire de files d'attente	chg voir les remarques 3 et 5	Non applicable
Changer de service	dsp et chg	Non applicable
Mettre à blanc une file d'attente	clr	Non applicable
Copier des informations d'authentification	dsp	crt
Copier les informations d'authentification (remplacement) voir la remarque 1	de: dsp à: chg	crt
Copier un canal	dsp	crt
Copier le canal (remplacer) voir la remarque 1	de: dsp à: chg	crt
Programme d'écoute de canal de copie	dsp	crt
Programme d'écoute de canal de copie (remplacement) voir la remarque 1	de: dsp à: chg	crt
Copier le canal de connexion client	dsp	crt
Copier le canal de connexion client (remplacement) voir la remarque 1	de: dsp à: chg	crt
Copier une liste de noms	dsp	crt

Tableau 2. Droits sur les objets (suite)

Commande	IBM MQ Les droits d'accès d'un objet	Droits sur les classes (pour le type d'objet)
Copier la liste de noms (remplacement) voir la remarque 1	de: dsp à: dsp et chg	crt
Copier un processus	dsp	crt
Processus de copie (remplacement) voir la remarque 1	de: dsp à: chg	crt
Copier une file d'attente	dsp	crt
File d'attente de copie (remplacement) voir la remarque 1	de: dsp à: dsp et chg	crt
Créer des informations d'authentification	(informations d'authentification par défaut du système) dsp	crt
Créer des informations d'authentification (remplacer) voir la remarque 1	(informations d'authentification par défaut du système) dsp sur: chg	crt
Créer un canal	(canal par défaut du système) dsp	crt
Créer un canal (remplacer) voir la remarque 1	(canal par défaut du système) dsp vers: chg	crt
Créer un programme d'écoute de canal	(programme d'écoute par défaut du système) dsp	crt
Créer un programme d'écoute de canal (remplacement) voir la remarque 1	(programme d'écoute par défaut du système) dsp sur: chg	crt
Créer un canal de connexion client	(canal par défaut du système) dsp	crt
Créer un canal de connexion client (remplacement) voir la remarque 1	(canal par défaut du système) dsp vers: chg	crt
Créer une liste de noms	(liste de noms par défaut du système) dsp	crt
Créer une liste de noms (remplacer) voir la remarque 1	(liste de noms par défaut du système) dsp sur: dsp et chg	crt
Créer un processus	(processus par défaut du système) dsp	crt
Créer un processus (remplacer) voir la remarque 1	(processus par défaut du système) dsp sur: chg	crt
Créer une file d'attente	(file d'attente par défaut du système) dsp	crt
Créer une file d'attente (remplacement) voir la remarque 1	(file d'attente par défaut du système) dsp vers: dsp et chg	crt
Créer un service	(file d'attente par défaut du système) dsp	crt

Tableau 2. Droits sur les objets (suite)

Commande	IBM MQ Les droits d'accès d'un objet	Droits sur les classes (pour le type d'objet)
Create Service (Replace) voir la remarque 1	(file d'attente par défaut du système) dsp vers: chg	crt
Supprimer des informations d'authentification	dsp et dlt	Non applicable
Supprimer l'enregistrement de droits d'accès	(objet gestionnaire de files d'attente) chg voir la remarque 4	voir la remarque 4
Supprimer un canal	dsp et dlt	Non applicable
Supprimer le programme d'écoute de canal	dsp et dlt	Non applicable
Supprimer le canal de connexion client	dsp et dlt	Non applicable
Supprimer une liste de noms	dsp et dlt	Non applicable
Supprimer un processus	dsp et dlt	Non applicable
Supprimer une file d'attente	dsp et dlt	Non applicable
Supprimer le service	dsp et dlt	Non applicable
Consulter des informations d'authentification	dsp	Non applicable
Consulter des enregistrements de droits	voir la remarque 4	voir la remarque 4
Consulter un canal	dsp	Non applicable
Consulter le programme d'écoute de canal	dsp	Non applicable
Interroger le statut du canal (pour <b>ChannelType</b> MQCHT_CLSSDR)	inq	Non applicable
Consulter le canal de connexion client	dsp	Non applicable
Consulter une liste de noms	dsp	Non applicable
Consulter un processus	dsp	Non applicable
Consulter la file d'attente	dsp	Non applicable
Consulter les gestionnaires de files d'attente	voir la remarque 3	Non applicable
Consulter le statut d'une file d'attente	dsp	Non applicable
Consulter un service	dsp	Non applicable
Envoyer une commande PING à un canal	ctrl	Non applicable
Envoyer une commande Ping à un gestionnaire de files d'attente	voir la remarque 3	Non applicable

Tableau 2. Droits sur les objets (suite)

Commande	IBM MQ Les droits d'accès d'un objet	Droits sur les classes (pour le type d'objet)
Régénérer un gestionnaire de files d'attente	(objet gestionnaire de files d'attente) chg	Non applicable
Actualiser la sécurité (pour <b>SecurityType</b> MQSECTYPE_SSL)	(objet gestionnaire de files d'attente) chg	Non applicable
Réinitialisation du canal	ctrlx	Non applicable
Réinitialiser un gestionnaire de files d'attente	(objet gestionnaire de files d'attente) chg	Non applicable
Réinitialiser les statistiques de file d'attente	dsp et chg	Non applicable
Résolution du canal	ctrlx	Non applicable
Définir l'enregistrement de droits d'accès	(objet gestionnaire de files d'attente) chg <i>voir la remarque 4</i>	<i>voir la remarque 4</i>
Démarrer un canal	ctrl	Non applicable
Arrêter le canal	ctrl	Non applicable
Arrêter une connexion	(objet gestionnaire de files d'attente) chg	Non applicable
Démarrage du programme d'écoute	ctrl	Non applicable
Arrêt du programme d'écoute	ctrl	Non applicable
Démarrer le service	ctrl	Non applicable
Arrêter le service	ctrl	Non applicable
Echap	<i>Voir la remarque 2</i>	<i>Voir la remarque 2</i>

**Remarques :**

1. Cette commande s'applique si l'objet à remplacer existe, sinon la vérification des droits est celle de la création ou de la copie sans remplacement.
2. Les droits requis sont déterminés par la commande MQSC définie par le texte d'échappement et sont équivalents à l'une des commandes précédentes.
3. Pour traiter une commande PCF, l'ID utilisateur doit disposer des droits dsp sur l'objet gestionnaire de files d'attente sur le système cible.
4. Cette commande PCF est autorisée sauf si le serveur de commandes a été démarré avec le paramètre -a. Par défaut, le serveur de commandes démarre lorsque le gestionnaire de files d'attente est démarré et sans le paramètre -a. Pour plus d'informations, voir [Programmable command formats reference](#).
5. L'octroi à un ID utilisateur de droits d'accès chg pour un gestionnaire de files d'attente permet de définir des enregistrements de droits d'accès pour tous les groupes et utilisateurs. N'accordez pas ce droit à des utilisateurs ou à des applications ordinaires.

IBM MQ fournit également des points d'exit de sécurité de canal afin que vous puissiez fournir vos propres programmes d'exit utilisateur pour le contrôle de la sécurité. Pour plus d'informations, voir [Affichage d'un canal](#).

## Utilisation de MQAI pour simplifier l'utilisation des fichiers PCF

L'interface d'administration IBM MQ (MQAI) est une interface de programmation d' IBM MQ disponible sur les systèmes AIX, IBM i, Linux, et Windows. Il effectue des tâches d'administration sur un gestionnaire de files d'attente IBM MQ à l'aide de sacs de données pour gérer les propriétés (ou les paramètres) des objets d'une manière plus simple que l'utilisation des formats PCF (Programmable Command Formats).

MQAI effectue des tâches d'administration sur un gestionnaire de files d'attente en utilisant des *sacs de données*. Les sacs de données vous permettent de gérer les propriétés (ou paramètres) des objets d'une manière plus simple que l'utilisation de fichiers PCF.

Les avantages de l'utilisation de MQAI sont les suivants:

### Simplifiez l'utilisation des messages PCF

Le MQAI est un moyen plus simple d'administrer IBM MQ. Si vous utilisez l'interface MQAI, vous n'avez pas besoin d'écrire vos propres messages PCF. Cela évite les problèmes liés aux structures de données complexes.

Pour transmettre des paramètres dans des programmes écrits à l'aide d'appels MQI, le message PCF doit contenir la commande et les détails de la chaîne ou des données entières. Pour créer cette configuration manuellement, vous devez ajouter plusieurs instructions dans votre programme pour chaque structure et vous devez allouer de l'espace mémoire. Cette tâche peut être longue et laborieuse.

Les programmes écrits à l'aide des MQAI passent des paramètres dans le sac de données approprié, et vous n'avez besoin que d'une seule instruction pour chaque structure. L'utilisation des sacs de données MQAI élimine la nécessité pour vous de gérer les grappes et d'allouer du stockage, et fournit un certain degré d'isolement par rapport aux détails du PCF.

### Gérer les conditions d'erreur plus facilement

Il est difficile d'obtenir des codes retour à partir des commandes PCF. L'interface MQAI permet au programme de gérer plus facilement les conditions d'erreur.

### Echange de données entre des applications

Les données d'application sont envoyées au format PCF et compressées et décompressées par MQAI. Si vos données de message se composent d'entiers et de chaînes de caractères, vous pouvez utiliser l'interface MQAI pour tirer parti de la conversion de données IBM MQ intégrée pour les données PCF. Cela évite d'avoir à écrire des exits de conversion de données.

Une fois que vous avez créé et rempli votre sac de données, vous pouvez envoyer un message de commande d'administration au serveur de commandes d'un gestionnaire de files d'attente à l'aide de l'appel `mqExecute`. Cet appel attend les messages de réponse. L'appel `mqExecute` gère l'échange avec le serveur de commandes et renvoie des réponses dans un *sac de réponse*.

## Exemples d'utilisation de MQAI

Les exemples de programme suivants illustrent l'utilisation de MQAI pour effectuer les différentes tâches:

- [amqsaicq.c](#): créez une file d'attente locale.
- [amqsaie.m](#): affiche les événements à l'écran à l'aide d'un moniteur d'événements simple.
- [amqsailq.c](#): permet d'imprimer la liste de toutes les files d'attente locales et de leurs profondeurs en cours.
- [amqsaicl.c](#): permet d'imprimer la liste de tous les canaux et de leurs types.

## Génération de votre application MQAI

Pour générer votre application à l'aide de MQAI, vous établissez un lien vers les mêmes bibliothèques que pour IBM MQ. Pour plus d'informations sur la génération de vos applications IBM MQ, voir [Génération d'une application procédurale](#).



## Conseils et astuces pour la configuration de IBM MQ à l'aide de MQAI

MQAI utilise des messages PCF pour envoyer des commandes d'administration au serveur de commandes au lieu de traiter directement avec le serveur de commandes lui-même. Des conseils pour la configuration de IBM MQ à l'aide de MQAI sont disponibles dans [«Conseils et astuces pour l'utilisation de MQAI afin de configurer IBM MQ»](#), à la page 41.

### Référence associée

[IBM MQ -Guide de référence de l'interface d'administration](#)

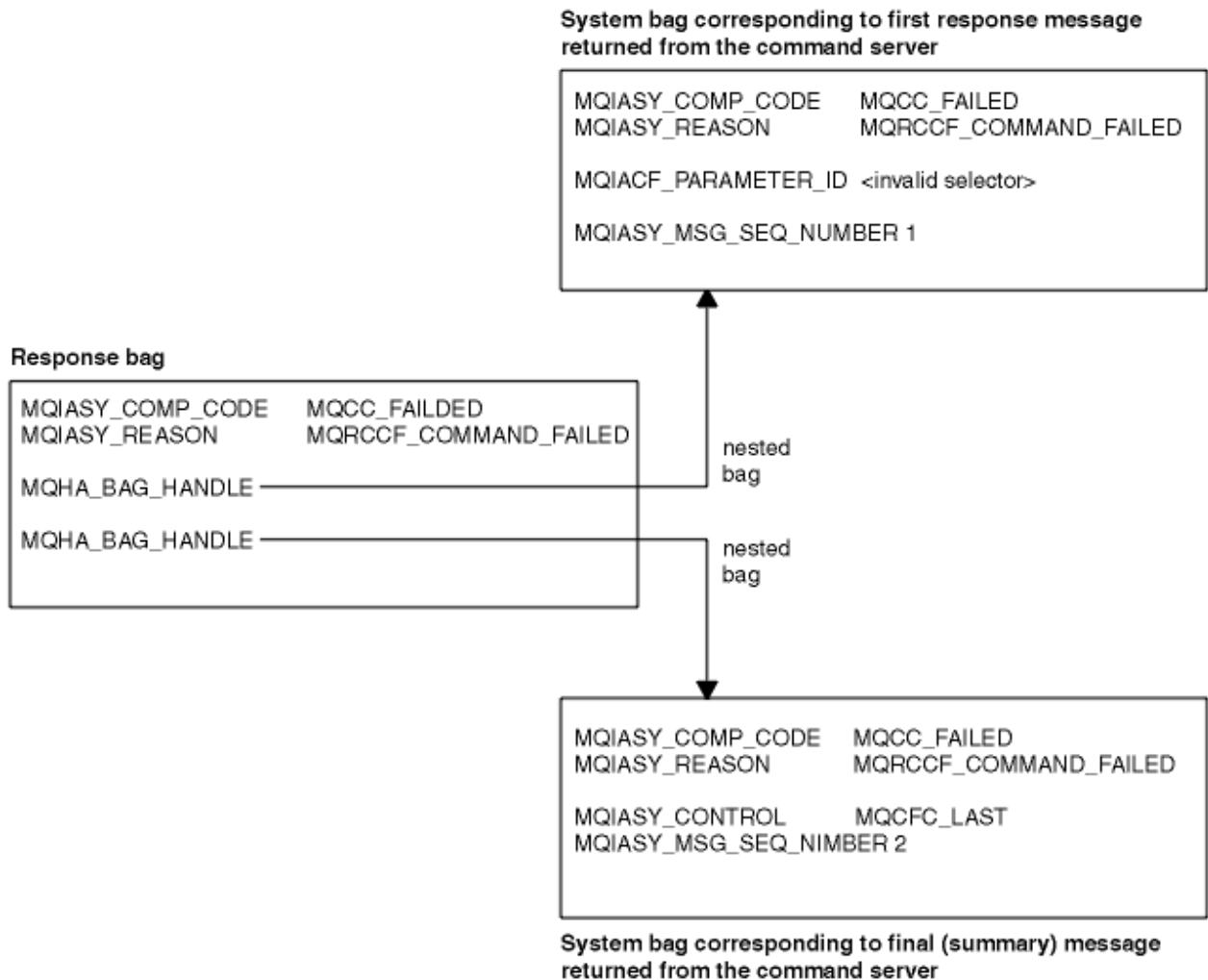
## **Multi** Conseils et astuces pour l'utilisation de MQAI afin de configurer IBM MQ

L'interface d'administration IBM MQ (MQAI) utilise des messages PCF pour envoyer des commandes d'administration au serveur de commandes plutôt que de traiter directement avec le serveur de commandes lui-même. Voici quelques conseils pour configurer IBM MQ à l'aide de MQAI.

- Les chaînes de caractères dans IBM MQ sont remplies de blancs à une longueur fixe. A l'aide de C, les chaînes à terminaison nulle peuvent normalement être fournies en tant que paramètres d'entrée aux interfaces de programmation IBM MQ .
- Pour effacer la valeur d'un attribut de chaîne, définissez-la sur un seul caractère blanc plutôt que sur une chaîne vide.
- Prenez en compte à l'avance les attributs que vous souhaitez modifier et n'examinez que ces attributs.
- Certains attributs ne peuvent pas être modifiés, par exemple un nom de file d'attente ou un type de canal. Veillez à ne modifier que les attributs qui peuvent être modifiés. Reportez-vous à la liste des paramètres obligatoires et facultatifs pour l'objet de modification PCF spécifique. Voir [Définitions des formats de commande programmables](#).
- Si un appel MQAI échoue, certains détails de l'échec sont renvoyés au sac de réponse. Des détails supplémentaires peuvent ensuite être trouvés dans un sac imbriqué accessible par le sélecteur MQHA\_BAG\_HANDLE. Par exemple, si un appel mqExecute échoue avec le code anomalie MQRCCF\_COMMAND\_FAILED, ces informations sont renvoyées dans le sac de réponse. Ce code raison peut être dû au fait qu'un sélecteur spécifié n'était pas valide pour le type de message de commande et que ce détail d'informations se trouve dans un sac imbriqué accessible par un descripteur de sac.

Pour plus d'informations sur mqExecute, voir [«Envoi de commandes d'administration au serveur de commandes qm à l'aide de l'appel mqExecute»](#), à la page 75

Le diagramme suivant illustre ce scénario:



## Multi Rubriques MQAI avancées

Informations sur l'indexation, la conversion de données et l'utilisation du descripteur de message

### Indexation

Les index sont utilisés lors du remplacement ou de la suppression d'éléments de données existants dans un sac afin de préserver l'ordre d'insertion.

### Conversion de données

Les chaînes contenues dans un sac de données MQAI peuvent se trouver dans divers jeux de caractères codés et peuvent être converties à l'aide de l'appel `mqSetInteger`.

### Utilisation du descripteur de message

Le MQAI génère un descripteur de message qui est défini sur une valeur initiale lors de la création du sac de données.

## Multi Indexation dans MQAI

Les index sont utilisés lors du remplacement ou de la suppression d'éléments de données existants dans un sac. Il existe trois types d'indexation, qui permettent d'extraire facilement des éléments de données.

Chaque sélecteur et valeur d'un élément de données dans un sac sont associés à trois numéros d'index:

- Index relatif à d'autres éléments ayant le même sélecteur.
- Index relatif à la catégorie du sélecteur (utilisateur ou système) à laquelle appartient l'élément.
- Index relatif à tous les éléments de données du sac (utilisateur et système).

Cela permet l'indexation par des sélecteurs d'utilisateur, des sélecteurs de système ou les deux, comme illustré dans la [Figure 3](#), à la page 43.

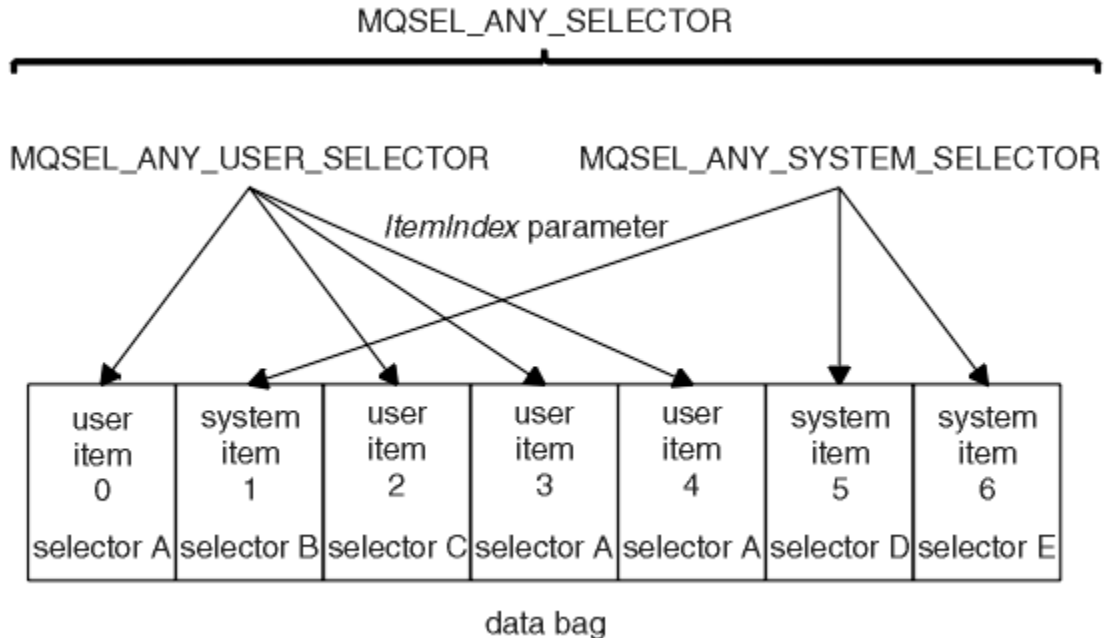


Figure 3. Indexation

Dans [Figure 3](#), à la page 43, l'élément utilisateur 3 (sélecteur A) peut être référencé par les paires d'index suivantes:

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

L'index est basé sur zéro comme un tableau en C ; s'il y a n'occurrences, l'index est compris entre zéro et 'n-1', sans écart.

Les index sont utilisés lors du remplacement ou de la suppression d'éléments de données existants dans un sac. Lorsqu'il est utilisé de cette manière, l'ordre d'insertion est conservé, mais les index d'autres éléments de données peuvent être affectés. Pour des exemples, voir «[Modification d'informations dans un sac](#)», à la page 72 et «[Suppression d'éléments de données](#)», à la page 74.

Les trois types d'indexation permettent d'extraire facilement des éléments de données. Par exemple, s'il existe trois instances d'un sélecteur particulier dans un sac, l'appel `mqCountItems` peut compter le nombre d'instances de ce sélecteur, et les appels `mqInquire*` peuvent spécifier à la fois le sélecteur et l'index pour interroger ces valeurs uniquement. Ceci est utile pour les attributs qui peuvent avoir une liste de valeurs telles que certaines des exits sur les canaux.

### **Multi** **Traitement de la conversion de données dans MQAI**

Les chaînes contenues dans un sac de données MQAI peuvent être dans une variété de jeux de caractères codés. Ces chaînes peuvent être converties à l'aide de l'appel `mqSetInteger`.

Comme les messages PCF, les chaînes contenues dans un sac de données MQAI peuvent être dans une variété de jeux de caractères codés. En général, toutes les chaînes d'un message PCF se trouvent dans le même jeu de caractères codés, c'est-à-dire dans le même jeu que le gestionnaire de files d'attente.

Chaque élément de chaîne d'un sac de données contient deux valeurs: la chaîne elle-même et le CCSID. La chaîne qui est ajoutée au sac est obtenue à partir du paramètre **Buffer** de l'appel `mqAddString` ou `mqSetString`. Le CCSID est obtenu à partir de l'élément système contenant un sélecteur de `MQIASY_CODED_CHAR_SET_ID`. Appelé *CCSID du sac*, il peut être modifié à l'aide de l'appel `mqSetInteger`.

Lorsque vous interrogez la valeur d'une chaîne contenue dans un sac de données, le CCSID est un paramètre de sortie de l'appel.

Le Tableau 3, à la page 44 montre les règles appliquées lors de la conversion de sacs de données en messages et vice versa:

<i>Tableau 3. Traitement CCSID</i>			
<b>Appel MQAI</b>	<b>CCSID</b>	<b>Entrée à appeler</b>	<b>Sortie à appeler</b>
<a href="#">mqBagToBuffer</a>	CCSID du sac ( <u>1</u> )	Ignorée	Inchangé
<a href="#">mqBagToBuffer</a>	CCSID de chaîne dans le sac	Utilisé	Inchangé
<a href="#">mqBagToBuffer</a>	CCSID de chaîne dans la mémoire tampon	Non applicable	Copié à partir des CCSID de chaîne dans le sac
<a href="#">mqBufferToBag</a>	CCSID du sac ( <u>1</u> )	Ignorée	Inchangé
<a href="#">mqBufferToBag</a>	CCSID de chaîne dans la mémoire tampon	Utilisé	Inchangé
<a href="#">mqBufferToBagmqBufferToBag</a>	CCSID de chaîne dans le sac	Non applicable	Copié à partir des CCSID de chaîne dans la mémoire tampon
<a href="#">SacmqPut</a>	CCSID MQMD	Utilisé	Inchangé ( <u>2</u> )
<a href="#">SacmqPut</a>	CCSID du sac ( <u>1</u> )	Ignorée	Inchangé
<a href="#">SacmqPut</a>	CCSID de chaîne dans le sac	Utilisé	Inchangé
<a href="#">SacmqPut</a>	CCSID de chaîne dans le message envoyé	Non applicable	Copié à partir des CCSID de chaîne dans le sac
<a href="#">mqGetSac</a>	CCSID MQMD	Utilisé pour la conversion de données de message	Défini sur le CCSID des données renvoyées ( <u>3</u> )
<a href="#">mqGetSac</a>	CCSID du sac ( <u>1</u> )	Ignorée	Inchangé
<a href="#">mqGetSac</a>	CCSID de chaîne dans le message	Utilisé	Inchangé
<a href="#">mqGetSac</a>	CCSID de chaîne dans le sac	Non applicable	Copié à partir des CCSID de chaîne dans le message
<a href="#">mqExecute</a>	CCSID du sac de demande	Utilisé pour le MQMD du message de demande ( <u>4</u> )	Inchangé
<a href="#">mqExecute</a>	CCSID du sac de réponse	Utilisé pour la conversion de données du message de réponse ( <u>4</u> )	Défini sur le CCSID des données renvoyées ( <u>3</u> )
<a href="#">mqExecute</a>	CCSID de chaîne dans le sac de demande	Utilisé pour le message de demande	Inchangé
<a href="#">mqExecute</a>	CCSID de chaîne dans le sac de réponse	Non applicable	Copié à partir des CCSID de chaîne dans le message de réponse

## Remarques :

1. Le CCSID du sac est l'élément système avec le sélecteur MQIASY\_CODED\_CHAR\_SET\_ID.
2. MQCCSI\_Q\_MGR est remplacé par le CCSID réel du gestionnaire de files d'attente.
3. Si une conversion de données est demandée, le CCSID des données renvoyées est identique à la valeur de sortie. Si la conversion de données n'est pas demandée, le CCSID des données renvoyées est identique à la valeur du message. Notez qu'aucun message n'est renvoyé si la conversion de données est demandée mais échoue.
4. Si le CCSID est MQCCSI\_DEFAULT, le CCSID du gestionnaire de files d'attente est utilisé.

## Concepts associés

«Conversion de données entre des jeux de caractères codés», à la page 214

Les données de message dans des formats définis par IBM MQ (également appelés formats intégrés) peuvent être converties par le gestionnaire de files d'attente d'un jeu de caractères codés à un autre, à condition que les deux jeux de caractères soient associés à une langue unique ou à un groupe de langues similaires.

«Le fichier ccsid\_part2.tbl», à la page 217

Le fichier ccsid\_part2.tbl est utilisé pour fournir des informations CCSID supplémentaires. Le fichier ccsid\_part2.tbl remplace le fichier ccsid.tbl utilisé avant IBM MQ 9.0.

### Multi

## Utilisation du descripteur de message dans MQAI

Le descripteur de message généré par MQAI est défini sur une valeur initiale lors de la création du sac de données.

Le type de commande PCF est obtenu à partir de l'élément système avec le sélecteur MQIASY\_TYPE. Lorsque vous créez votre sac de données, la valeur initiale de cet élément est définie en fonction du type de sac que vous créez:

Type de sac	Valeur initiale de l'élément MQIASY_TYPE
MQCBO_ADMIN_BAG	MQCFT_COMMAND (COMMANDE MQ)
SAC de commande mqcbo_commande	MQCFT_COMMAND (COMMANDE MQ)
MQCBO_*	Utilisateur_MQCF

Lorsque MQAI génère un descripteur de message, les valeurs utilisées dans les paramètres **Format** et **MsgType** dépendent de la valeur de l'élément système avec le sélecteur MQIASY\_TYPE, comme illustré dans la [Tableau 4](#), à la page 45.

Type de commande PCF	Format	MsgType
MQCFT_COMMAND (COMMANDE MQ)	MQFMT_ADMIN	MQMT_REQUEST
RAPPORT MQCF	MQFMT_ADMIN	MQMT_REPORT
MQCFT_REPONSE	MQFMT_ADMIN	MQMT_REPLY
MQCFT_TRACE_ROUTE	MQFMT_ADMIN	MQMT_DATAGRAM
MQCFT_EVENT	MQFMT_EVENT	MQMT_DATAGRAM
MQCFT_*	MQFMT_PCF	MQMT_DATAGRAM

Tableau 5, à la page 45 montre que si vous créez un sac d'administration ou un sac de commande, le *Format* du descripteur de message est MQFMT\_ADMIN et le *MsgType* est MQMT\_REQUEST. Cela

convient pour un message de demande PCF envoyé au serveur de commandes lorsqu'une réponse est attendue.

Les autres paramètres du descripteur de message prennent les valeurs indiquées dans le [Tableau 6](#), à la page 46.

<i>Tableau 6. Valeurs de descripteur de message</i>	
<b>Paramètre</b>	<b>Valeur</b>
<i>StrucId</i>	ID_STRUCD_MQM
<i>Version</i>	MQMD_VERSION_1
<i>Report</i>	MQRO_AUCUN
<i>MsgType</i>	Voir <a href="#">Tableau 5</a> , à la page 45
<i>Expiry</i>	30 secondes (remarque «1», à la page 46)
<i>Feedback</i>	MQFB_AUCUN
<i>Encoding</i>	MQENC_NATIVE
<i>CodedCharSetId</i>	dépend du CCSID du sac (remarque «2», à la page 46)
<i>Format</i>	Voir <a href="#">Tableau 5</a> , à la page 45
<i>Priority</i>	MQPRI_PRIORITY_AS_Q_DEF
<i>Persistence</i>	MQPER_NON_PERSISTENT
<i>MsgId</i>	MQMI_AUCUN
<i>CorrelId</i>	MQCI_NONE
<i>BackoutCount</i>	0
<i>ReplyToQ</i>	Voir la remarque «3», à la page 46
<i>ReplyToQMgr</i>	blanc

#### Remarques :

1. Cette valeur peut être remplacée dans l'appel `mqExecute` à l'aide du paramètre **OptionsBag**. Pour plus d'informations, voir `mqExecute`.
2. Voir «[Traitement de la conversion de données dans MQAI](#)», à la page 43.
3. Nom de la file d'attente de réponses spécifiée par l'utilisateur ou de la file d'attente dynamique temporaire générée par MQAI pour les messages de type MQMT\_REQUEST. Vide dans le cas contraire.

### **Multi** Exemple de programme C pour la création d'une file d'attente locale (amqsaicq.c)

L'exemple de programme C `amqsaicq.c` crée une file d'attente locale à l'aide de MQAI.

```

/*****
/*
/* Program name: AMQSAICQ.C
/*
/* Description: Sample C program to create a local queue using the
/* IBM MQ Administration Interface (MQAI).
/*
/* Statement: Licensed Materials - Property of IBM
/*
/* 84H2000, 5765-B73
/* 84H2001, 5639-B42
/* 84H2002, 5765-B74
*/

```

```

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

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

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

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

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

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

    /*****
    /* Report reason and stop if connection failed
    /*****
    if (compCode == MQCC_FAILED)

```

```

    {
        CheckCallResult("MQCONN", compCode, connReason);
        exit( (int)connReason);
    }

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

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

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

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

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

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

```



```

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

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

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

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

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

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

/*****
/* Delete the command bag if successfully created. */
/*****
if (commandBag != MQHB_UNUSABLE_HBAG)
{
    mqDeleteBag(&commandBag, &compCode, &reason);
}

```

```

    CheckCallResult("Delete the command bag", compCode, reason);
}

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

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

```

## Multi Exemple de programme C pour l'affichage d'événements à l'aide d'un moniteur d'événements (amqsaiem.c)

L'exemple de programme C amqsaiem.c illustre un moniteur d'événements de base à l'aide de MQAI.

```

*****/
/*
/* Program name: AMQSAIEM.C
/*
/* Description: Sample C program to demonstrate a basic event monitor
/* using the IBM MQ Admin Interface (MQAI).
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1999, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
*****/
/*
/* Function:
/* AMQSAIEM is a sample C program that demonstrates how to write a simple
/* event monitor using the mqGetBag call and other MQAI calls.
/*
/* The name of the event queue to be monitored is passed as a parameter
/* to the program. This would usually be one of the system event queues:-
/* SYSTEM.ADMIN.QMGR.EVENT Queue Manager events
/* SYSTEM.ADMIN.PERFM.EVENT Performance events
/* SYSTEM.ADMIN.CHANNEL.EVENT Channel events
/* SYSTEM.ADMIN.LOGGER.EVENT Logger events
/*
/* To monitor the queue manager event queue or the performance event queue,
/* the attributes of the queue manager need to be changed to enable
/* these events. For more information about this, see Part 1 of the
/* Programmable System Management book. The queue manager attributes can
/* be changed using either MQSC commands or the MQAI interface.
/* Channel events are enabled by default.
*****/

```

```

/* Program logic
/* Connect to the Queue Manager.
/* Open the requested event queue with a wait interval of 30 seconds.
/* Wait for a message, and when it arrives get the message from the queue
/* and format it into an MQAI bag using the mqGetBag call.
/* There are many types of event messages and it is beyond the scope of
/* this sample to program for all event messages. Instead the program
/* prints out the contents of the formatted bag.
/* Loop around to wait for another message until either there is an error
/* or the wait interval of 30 seconds is reached.
/*
/*****
/*
/* AMQSAIEM has 2 parameters - the name of the event queue to be monitored
/* - the queue manager name (optional)
/*
/*****

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

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

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

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

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

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

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

```

```

        exit( (int)connReason);
    }

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

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

    return 0;
}

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

/*****
/*
/* Function: GetQEvents
/*
/*
*****/
/*
/* Input Parameters: Handle to the queue manager
/* Name of the event queue to be monitored
/*
/*
/* Output Parameters: None
/*
/*
/* Logic: Open the event queue.
/* Get a message off the event queue and format the message into
/* a bag.
/* A real event monitor would need to be programmed to deal with
/* each type of event that it receives from the queue. This is
/* outside the scope of this sample, so instead, the contents of
/* the bag are printed.
/* The program waits for 30 seconds for an event message and then
/* terminates if no more messages are available.
*****/
void GetQEvents(MQHCONN hConn, MQCHAR *qName)
{
    MQLONG openReason; /* MQOPEN reason code */
    MQLONG reason; /* reason code */
    MQLONG compCode; /* completion code */
    MQHOBJ eventQueue; /* handle to event queue */

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

    /*****/

```

```

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

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

/*****
/* Set the GMO options to control the action of the get message from the */
/* queue. */
/*****
gmo.WaitInterval = 30000; /* 30 second wait for message */
gmo.Options = MQGMO_WAIT + MQGMO_FAIL_IF_QUIESCING + MQGMO_CONVERT;
gmo.Version = MQGMO_VERSION 2; /* Avoid need to reset Message ID */
gmo.MatchOptions = MQMO_NONE; /* and Correlation ID after every */
/* mqGetBag

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

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

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

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

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

/*****
/* Event message read - Print the contents of the event bag */
/*****
else
{
    if ( PrintBag(eventBag) )
        printf("\nError found while printing bag contents\n");
} /* end of msg found */
} /* end of main loop */
/*****
/* Close the event queue if successfully opened */
/*****
if (openReason == MQRC_NONE)
{
    MQCLOSE(hConn, &eventQueue, MQCO_NONE, &compCode, &reason);
    CheckCallResult("Close event queue", compCode, reason);

```

```

}

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

} /* end of GetQEvents */

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

int PrintBag(MQHBAG dataBag)
{
    int errors;

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

    return errors;
}

/*****
/*
/* Function: PrintBagContents
/*
/*
/* Input Parameters: Bag Handle
/* Indentation level of bag
/*
/* Output Parameters: None
/*
/* Returns: Number of errors found
/*
/* Logic: Count the number of items in the bag
/* Obtain selector and item type for each item in the bag.
/* Obtain the value of the item depending on item type and display the
/* index of the item, the selector and the value.
/* If the item is an embedded bag handle then call this function again
/* to print the contents of the embedded bag increasing the
/* indentation level.
/*
/*
/*****
int PrintBagContents(MQHBAG dataBag, int indent)
{
    /*****
    /* Definitions
    /*****
    #define LENGTH 500 /* Max length of string to be read*/
    #define INDENT 4 /* Number of spaces to indent */
    /* embedded bag display */

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

```

```

MQINT32 iValue; /* Value if item is an integer */
MQINT64 i64Value; /* Value if item is a 64-bit integer */
MQLONG selector; /* Selector of item */
MQHBAG bagHandle; /* Value if item is a bag handle */
MQLONG reason; /* reason code */
MQLONG compCode; /* completion code */
MQLONG trimLength; /* Length of string to be trimmed */
int errors = 0; /* Count of errors found */
char blanks[] = " "; /* Blank string used to indent display */

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

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

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

        if (compCode != MQCC_OK)
            errors++;

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

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

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

```

```

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

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

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

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

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

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

            printf("' \n");
    }
    break;

```



```

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

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

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

```

## Multi Exemple de programme C pour l'interrogation d'objets de canal (amqsaicl.c)

L'exemple de programme C amqsaicl.c demande des objets de canal à l'aide de MQAI.

```

/******
/*
/* Program name: AMQSAICL.C
/*
/* Description: Sample C program to inquire channel objects
/* using the IBM MQ Administration Interface (MQAI)
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 2008, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/******
/*
/* Function:
/* AMQSAICL is a sample C program that demonstrates how to inquire
/* attributes of the local queue manager using the MQAI interface. In
/* particular, it inquires all channels and their types.
/*
/* - A PCF command is built from items placed into an MQAI administration
/* bag.
/* These are:-
/* - The generic channel name "*"
/* - The attributes to be inquired. In this sample we just want
/* name and type attributes
/*
/* - The mqExecute MQCMD_INQUIRE_CHANNEL call is executed.
/* The call generates the correct PCF structure.
/* The default options to the call are used so that the command is sent
/* to the SYSTEM.ADMIN.COMMAND.QUEUE.
/* The reply from the command server is placed on a temporary dynamic
/* queue.
/*

```

```

/*      The reply from the MQCMD_INQUIRE_CHANNEL is read from the      */
/*      temporary queue and formatted into the response bag.            */
/*      */
/*      - The completion code from the mqExecute call is checked and if there */
/*      is a failure from the command server, then the code returned by the */
/*      command server is retrieved from the system bag that has been */
/*      embedded in the response bag to the mqExecute call.            */
/*      */
/* Note: The command server must be running.                            */
/*      */
/*****
/* AMQSAICL has 2 parameter - the queue manager name (optional)        */
/*                          - output file (optional) default varies    */
*****/

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

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

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

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

/*****
/* Constants                                                          */
*****/
#if (MQAT_DEFAULT == MQAT_OS400)
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    " *SDR      ", /* MQCHT_SENDER      */
    " *SVR      ", /* MQCHT_SERVER      */
    " *RCVR     ", /* MQCHT_RECEIVER   */
    " *RQSTR    ", /* MQCHT_REQUESTER  */
    " *ALL      ", /* MQCHT_ALL         */
    " *CLTCN    ", /* MQCHT_CLNTCONN   */
    " *SVRCONN  ", /* MQCHT_SVRCONN    */
    " *CLUSRCVR", /* MQCHT_CLUSRCVR   */
    " *CLUSSDR  ", /* MQCHT_CLUSSDR    */
};
#else
const struct
{
    char name[9];
} ChlTypeMap[9] =
{
    "sdr      ", /* MQCHT_SENDER      */
    "svr      ", /* MQCHT_SERVER      */
    "rcvr     ", /* MQCHT_RECEIVER   */
    "rqstr    ", /* MQCHT_REQUESTER  */
    "all      ", /* MQCHT_ALL         */
    "cltconn  ", /* MQCHT_CLNTCONN   */
    "svrcn   ", /* MQCHT_SVRCONN    */
    "clusrcvr", /* MQCHT_CLUSRCVR   */
    "clussdr  ", /* MQCHT_CLUSSDR    */
};
#endif

```

```

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

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

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

#endif

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

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

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

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

/*****
/* Open the output file
/*****
if (argc > 2)

```

```

}
OPENOUTFILE(outfp, argv[2]);
}
else
{
OPENOUTFILE(outfp, OUTFILE);
}

if(outfp == NULL)
{
printf("Could not open output file.\n");
goto MOD_EXIT;
}
/*****
/* Create an admin bag for the mqExecute call */
/*****
mqCreateBag(MQCBO_ADMIN_BAG, &adminBag;, &compCode;, &reason;);
CheckCallResult("Create admin bag", compCode, reason);

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

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

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

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

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

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

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

```

```

CheckCallResult("Count number of bag handles", compCode, reason);

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

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

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

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

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

    /*
    /* Get the completion code and reason code, returned by the command
    /* server, from the embedded error bag.
    /*
    /*
    mqInquireInteger(errorBag, MQIASY_COMP_CODE, MQIND_NONE, &mqExecuteCC,
      &compCode, &reason );
    CheckCallResult("Get the completion code from the result bag",
      compCode, reason);
    mqInquireInteger(errorBag, MQIASY_REASON, MQIND_NONE, &mqExecuteRC,
      &compCode, &reason);
    CheckCallResult("Get the reason code from the result bag",
      compCode, reason);
    printf("Error returned by the command server: Cc = %ld : Rc = %ld\n",
      mqExecuteCC, mqExecuteRC);
  }
}

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

/*
/* Delete the response bag if successfully created.
/*
/*

```

```

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

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

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

return 0;
}

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

```

## Exemple de programme C pour l'interrogation des files d'attente et des informations d'impression (amqsailq.c)

L'exemple de programme C amqsailq.c demande la longueur en cours des files d'attente locales à l'aide de MQAI.

```

/*****
/*
/* Program name: AMQSAILQ.C */
/*
/* Description: Sample C program to inquire the current depth of the local */
/* queues using the IBM MQ Administration Interface (MQAI) */
/*
/* Statement: Licensed Materials - Property of IBM */
/*
/* 84H2000, 5765-B73 */
/* 84H2001, 5639-B42 */
/* 84H2002, 5765-B74 */
/* 84H2003, 5765-B75 */
/* 84H2004, 5639-B43 */
/*
/* (C) Copyright IBM Corp. 1999, 2024 */
/*
/*****
/*
/* Function: */
/* AMQSAILQ is a sample C program that demonstrates how to inquire */
/* attributes of the local queue manager using the MQAI interface. In */
/* particular, it inquires the current depths of all the local queues. */

```

```

/*
/* - A PCF command is built by placing items into an MQAI administration */
/* bag. */
/* These are:- */
/* - The generic queue name "*" */
/* - The type of queue required. In this sample we want to */
/* inquire local queues. */
/* - The attribute to be inquired. In this sample we want the */
/* current depths. */
/*
/* - The mqExecute call is executed with the command MQCMD_INQUIRE_Q. */
/* The call generates the correct PCF structure. */
/* The default options to the call are used so that the command is sent */
/* to the SYSTEM.ADMIN.COMMAND.QUEUE. */
/* The reply from the command server is placed on a temporary dynamic */
/* queue. */
/* The reply from the MQCMD_INQUIRE_Q command is read from the */
/* temporary queue and formatted into the response bag. */
/*
/* - The completion code from the mqExecute call is checked and if there */
/* is a failure from the command server, then the code returned by */
/* command server is retrieved from the system bag that has been */
/* embedded in the response bag to the mqExecute call. */
/*
/* - If the call is successful, the depth of each local queue is placed */
/* in system bags embedded in the response bag of the mqExecute call. */
/* The name and depth of each queue is obtained from each of the bags */
/* and the result displayed on the screen. */
/*
/* Note: The command server must be running. */
/*
/*****
/* AMQSAILQ has 1 parameter - the queue manager name (optional) */
/*
/*****

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

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

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

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

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

```

```

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

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

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

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

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

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

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

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

/*****
/* Check the result from mqExecute call. If successful find the current */
/* depths of all the local queues. If failed find the error. */
/*****
if ( compCode == MQCC_OK ) /* Successful mqExecute */
{
    /*****

```



```

/* Count the number of system bags embedded in the response bag from the */
/* mqExecute call. The attributes for each queue are in a separate bag. */
/*****
mqCountItems(responseBag, MQHA_BAG_HANDLE, &numberOfBags, &compCode,
              &reason);
CheckCallResult("Count number of bag handles", compCode, reason);

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

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

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

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

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

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

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

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

```

```

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

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

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

```

## Multi Sacs de données et MQAI

Un sac de données est un moyen de gérer les propriétés ou les paramètres des objets à l'aide de l'interface d'administration IBM MQ (MQAI).

### Sacs de données

- Le sac de données contient zéro ou plusieurs *éléments de données*. Ces éléments de données sont commandés dans le sac au fur et à mesure qu'ils sont placés dans le sac. Il s'agit de l' *ordre d'insertion*. Chaque élément de données contient un *sélecteur* qui identifie l'élément de données et une *valeur* de cet élément de données qui peut être un entier, un entier 64 bits, un filtre d'entier, une chaîne, un filtre de chaîne, une chaîne d'octets, un filtre de chaîne d'octets ou un descripteur d'un autre sac. Les éléments de données sont décrits en détail dans «Types d'élément de données disponibles dans MQAI», à la page 69

Il existe deux types de sélecteur: *sélecteurs d'utilisateur* et *sélecteurs de système*. Ces éléments sont décrits dans Sélecteurs MQAI. Les sélecteurs sont généralement uniques, mais il est possible d'avoir plusieurs valeurs pour le même sélecteur. Dans ce cas, un *index* identifie l'occurrence particulière du sélecteur requise. Les index sont décrits dans «Indexation dans MQAI», à la page 42.

Une hiérarchie de ces concepts est illustrée dans la [Figure 1](#).

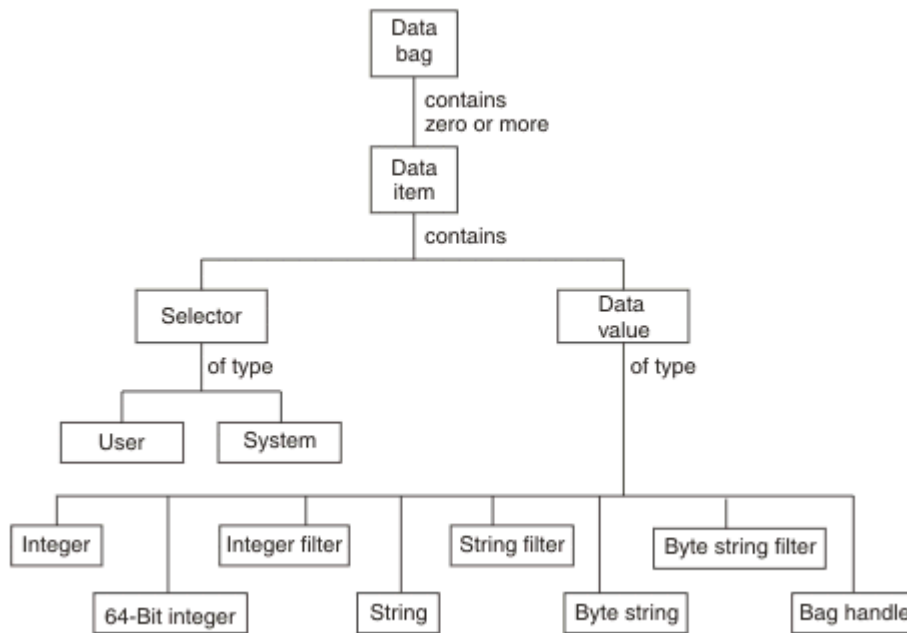


Figure 4. Hiérarchie des concepts MQAI

La hiérarchie a été expliquée dans un paragraphe précédent.

## Types de sac de données

Vous pouvez choisir le type de sac de données que vous souhaitez créer en fonction de la tâche que vous souhaitez effectuer:

### sac utilisateur

Sac simple utilisé pour les données utilisateur.

### jeu de données d'administration

Sac créé pour les données utilisées pour administrer les objets IBM MQ en envoyant des messages d'administration à un serveur de commandes. Le sac d'administration implique automatiquement certaines options, comme décrit dans [«Création et suppression de sacs de données»](#), à la page 68.

### sac de commandes

Un sac a également été créé pour les commandes d'administration des objets IBM MQ . Cependant, contrairement au sac d'administration, le sac de commande n'implique pas automatiquement certaines options bien que ces options soient disponibles. Pour plus d'informations sur les options, voir [«Création et suppression de sacs de données»](#), à la page 68.

### sac de groupe

Sac utilisé pour contenir un ensemble d'éléments de données groupés. Les sacs de groupe ne peuvent pas être utilisés pour l'administration des objets IBM MQ .

En outre, le **sac système** est créé par MQAI lorsqu'un message de réponse est renvoyé par le serveur de commandes et placé dans le sac de sortie d'un utilisateur. Un sac système ne peut pas être modifié par l'utilisateur.

Utilisation des sacs de données Les différentes façons d'utiliser les sacs de données sont répertoriées dans cette rubrique:

## Utilisation de sacs de données

Les différentes méthodes d'utilisation des sacs de données sont présentées dans la liste suivante:

- Vous pouvez créer et supprimer des sacs de données [«Création et suppression de sacs de données»](#), à la page 68.

- Vous pouvez envoyer des données entre des applications à l'aide de sacs de données [«Insertion et réception de sacs de données à l'aide de MQAI»](#), à la page 69.
- Vous pouvez ajouter des éléments de données à des sacs de données [«Ajout d'éléments de données à des sacs avec MQAI»](#), à la page 70.
- Vous pouvez ajouter une commande d'interrogation dans un sac de données [«Ajout d'une instruction d'interrogation à un sac»](#), à la page 71.
- Vous pouvez vous renseigner dans les sacs de données [«Recherche dans les sacs de données»](#), à la page 71.
- Vous pouvez compter des éléments de données dans un sac de données [«Comptage des éléments de données»](#), à la page 74.
- Vous pouvez modifier des informations dans un sac de données [«Modification d'informations dans un sac»](#), à la page 72.
- Vous pouvez effacer un sac de données [«Effacement d'un sac à l'aide de l'appel de sac mqClear»](#), à la page 73.
- Vous pouvez tronquer un sac de données [«Troncation d'un sac à l'aide de l'appel de sac mqTruncate»](#), à la page 73.
- Vous pouvez convertir des sacs et des tampons [«Conversion de sacs et de tampons»](#), à la page 73.

## **Multi** *Création et suppression de sacs de données*

### **Création de sacs de données**

Pour utiliser MQAI, vous devez d'abord créer un sac de données à l'aide de l'appel mqCreateBag. En entrée de cet appel, vous fournissez une ou plusieurs options pour contrôler la création du sac.

Le paramètre **Options** de l'appel MQCreateBag vous permet de choisir de créer un sac d'utilisateur, un sac de commande, un sac de groupe ou un sac d'administration.

Pour créer un sac d'utilisateur, un sac de commande ou un sac de groupe, vous pouvez choisir une ou plusieurs options supplémentaires pour:

- Utilisez le formulaire de liste lorsqu'il y a plusieurs occurrences adjacentes du même sélecteur dans un sac.
- Réorganisez les éléments de données au fur et à mesure qu'ils sont ajoutés à un message PCF pour vous assurer que les paramètres sont dans le bon ordre. Pour plus d'informations sur les éléments de données, voir [«Types d'élément de données disponibles dans MQAI»](#), à la page 69.
- Vérifiez les valeurs des sélecteurs d'utilisateur pour les éléments que vous ajoutez au sac.

Les sacs d'administration impliquent automatiquement ces options.

Un sac de données est identifié par sa poignée. Le descripteur de sac est renvoyé par le sac mqCreateet doit être fourni sur tous les autres appels qui utilisent le sac de données.

Pour une description complète de l'appel de sac mqCreate, voir [mqCreateBag](#).

### **Suppression de sacs de données**

Tout sac de données créé par l'utilisateur doit également être supprimé à l'aide de l'appel de sac mqDelete. Par exemple, si un sac est créé dans le code utilisateur, il doit également être supprimé dans le code utilisateur.

Les sacs système sont créés et supprimés automatiquement par MQAI. Pour plus d'informations à ce sujet, voir [«Envoi de commandes d'administration au serveur de commandes qm à l'aide de l'appel mqExecute»](#), à la page 75. Le code utilisateur ne peut pas supprimer un sac système.

Pour une description complète de l'appel de sac mqDelete, voir [mqDeleteBag](#).

## ***Insertion et réception de sacs de données à l'aide de MQAI***

Les données peuvent également être envoyées entre les applications en plaçant et en obtenant des sacs de données à l'aide des appels de sac mqPutet mqGet. Cela permet à l'interface d'administration IBM MQ (MQAI) de gérer la mémoire tampon plutôt que l'application.

L'appel de sac mqPutconvertit le contenu du sac spécifié en message PCF et envoie le message à la file d'attente spécifiée ; l'appel de sac mqGetsupprime le message de la file d'attente spécifiée et le reconvertit en sac de données. Par conséquent, l'appel mqPutest l'équivalent de l'appel mqBagToBuffer suivi de MQPUT, et l'appel mqGetBag est l'équivalent de l'appel MQGET suivi de mqBufferToBag.

Pour plus d'informations sur l'envoi et la réception de messages PCF dans une file d'attente spécifique, voir [«Envoi et réception de messages PCF dans une file d'attente spécifiée», à la page 30](#)

**Remarque :** Si vous choisissez d'utiliser l'appel mqGetBag, les détails PCF dans le message doivent être corrects ; si ce n'est pas le cas, une erreur appropriée se produit et le message PCF n'est pas renvoyé.

## ***Types d'élément de données disponibles dans MQAI***

Les éléments de données sont utilisés par l'interface d'administration IBM MQ (MQAI) pour remplir les sacs de données lorsqu'ils sont créés. Ces éléments de données peuvent être des éléments utilisateur ou système.

Ces éléments utilisateur contiennent des données utilisateur telles que les attributs des objets en cours de gestion. Les éléments système doivent être utilisés pour mieux contrôler les messages générés: par exemple, la génération d'en-têtes de message. Pour plus d'informations sur les éléments système, voir [«Éléments système et MQAI», à la page 69](#).

### **Types d'éléments de données**

Une fois que vous avez créé un sac de données, vous pouvez le remplir avec des éléments de type entier ou chaîne de caractères. Vous pouvez vous renseigner sur les trois types d'élément.

L'élément de données peut être un entier ou une chaîne de caractères. Voici les types d'élément de données disponibles dans MQAI:

- Entier
- Entier 64 bits
- Filtre de type entier
- Chaîne de caractères
- Filtre de chaîne
- Chaîne d'octets
- Filtre de chaîne d'octets
- Poignée de sac

### **Utilisation d'éléments de données**

Les méthodes d'utilisation des éléments de données sont les suivantes:

- [«Comptage des éléments de données», à la page 74](#).
- [«Suppression d'éléments de données», à la page 74](#).
- [«Ajout d'éléments de données à des sacs avec MQAI», à la page 70](#).
- [«Filtrage et interrogation des éléments de données», à la page 71](#).

## ***Éléments système et MQAI***

Les éléments système peuvent être utilisés par l'interface d'administration IBM MQ (MQAI) pour:

- Génération des en-têtes PCF. Les éléments système peuvent contrôler l'identificateur de commande PCF, les options de contrôle, le numéro de séquence de message et le type de commande.

- Conversion de données. Les éléments système gèrent l'identificateur de jeu de caractères pour les éléments de chaîne de caractères du sac.

Comme tous les éléments de données, les éléments système sont constitués d'un sélecteur et d'une valeur. Pour plus d'informations sur ces sélecteurs et sur leur fonction, voir [Sélecteurs MQAI](#).

Les éléments système sont uniques. Un ou plusieurs éléments système peuvent être identifiés par un sélecteur de système. Il n'y a qu'une seule occurrence de chaque sélecteur de système.

La plupart des éléments système peuvent être modifiés (voir «[Modification d'informations dans un sac](#)», à la page 72), mais les options de création de sac ne peuvent pas être modifiées par l'utilisateur. Vous ne pouvez pas supprimer d'éléments système. (Voir «[Suppression d'éléments de données](#)», à la page 74.)

#### **Multi** Ajout d'éléments de données à des sacs avec MQAI

Lorsqu'un sac de données est créé à l'aide de l'interface d'administration IBM MQ (MQAI), vous pouvez le remplir avec des éléments de données. Ces éléments de données peuvent être des éléments utilisateur ou système.

Pour plus d'informations sur les éléments de données, voir «[Types d'élément de données disponibles dans MQAI](#)», à la page 69.

L'interface MQAI vous permet d'ajouter des éléments de type entier, des éléments de type entier 64 bits, des éléments de type filtre de type entier, des éléments de type chaîne de caractères, des éléments de type chaîne de caractères, des éléments de type chaîne d'octets et des éléments de type filtre de type chaîne d'octets à des sacs, comme indiqué dans la [Figure 5](#), à la page 70. Les éléments sont identifiés par un sélecteur. Généralement, un sélecteur identifie un seul élément, mais ce n'est pas toujours le cas. Si un élément de données avec le sélecteur spécifié est déjà présent dans le sac, une instance supplémentaire de ce sélecteur est ajoutée à la fin du sac.

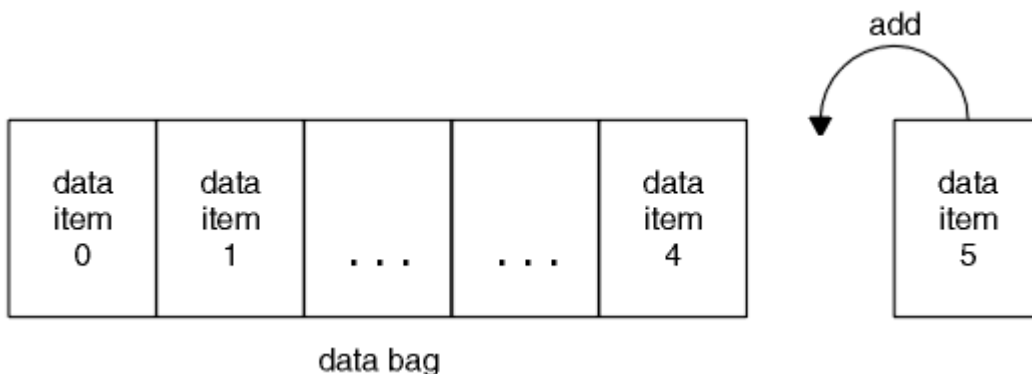


Figure 5. Ajout d'éléments de données

Ajoutez des éléments de données à un sac à l'aide des appels mqAdd\*:

- Pour ajouter des éléments de type entier, utilisez l'appel [mqAddInteger](#) comme décrit dans [mqAddInteger](#)
- Pour ajouter des entiers 64 bits, utilisez l'appel [mqAddInteger64](#) comme décrit dans [mqAddInteger64](#)
- Pour ajouter des éléments de filtre de type entier, utilisez l'appel [mqAddIntegerFilter](#) comme décrit dans [mqAddIntegerFilter](#)
- Pour ajouter des éléments de chaîne de caractères, utilisez l'appel de chaîne [mqAddString](#) comme décrit dans [mqAddString](#)
- Pour ajouter des éléments de filtre de chaîne, utilisez l'appel [mqAddStringFilter](#) comme décrit dans [mqAddStringFilter](#)
- Pour ajouter des éléments de chaîne d'octets, utilisez l'appel [mqAddByteString](#) comme décrit dans [mqAddByteString](#)
- Pour ajouter des éléments de filtre de chaîne d'octets, utilisez l'appel de filtre [mqAddByteStringFilter](#) comme décrit dans [mqAddByteStringFilter](#)

Pour plus d'informations sur l'ajout d'éléments de données à un sac, voir «[Eléments système et MQAI](#)», à la page 69.

#### **Multi** Ajout d'une instruction d'interrogation à un sac

L'appel d'interrogation `mqAdd` permet d'ajouter une commande d'interrogation à un sac. L'appel est spécifique à des fins d'administration, de sorte qu'il peut être utilisé avec des sacs d'administration uniquement. Il vous permet de spécifier les sélecteurs d'attributs sur lesquels vous souhaitez vous renseigner à partir de IBM MQ.

Pour une description complète de l'appel `mqAddInquiry`, voir [mqAddInquiry](#).

#### **Multi** Filtrage et interrogation des éléments de données

Lorsque vous utilisez MQAI pour vous renseigner sur les attributs des objets IBM MQ, vous pouvez contrôler les données renvoyées à votre programme de deux manières.

- Vous pouvez **filtrer** les données renvoyées à l'aide des appels de chaîne `mqAddInteger` et `mqAdd`. Cette approche vous permet de spécifier une paire *Selector* et *ItemValue*, par exemple:

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

Cet exemple indique que le type de file d'attente (*Selector*) doit être local (*ItemValue*) et que cette spécification doit correspondre aux attributs de l'objet (dans ce cas, une file d'attente) que vous interrogez.

Les autres attributs qui peuvent être filtrés correspondent aux commandes PCF Inquire \* qui se trouvent dans «[Introduction à IBM MQ Programmable Command Formats](#)», à la page 27. Par exemple, pour en savoir plus sur les attributs d'un canal, reportez-vous à la commande Inquire Channel dans la documentation du produit. Les "paramètres requis" et les "paramètres facultatifs" de la commande Inquire Channel identifient les sélecteurs que vous pouvez utiliser pour le filtrage.

- Vous pouvez **interroger** les attributs particuliers d'un objet à l'aide de l'appel d'interrogation `mqAdd`. Indique le sélecteur qui vous intéresse. Si vous ne spécifiez pas le sélecteur, tous les attributs de l'objet sont renvoyés.

Voici un exemple de filtrage et d'interrogation des attributs d'une file d'attente:

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

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

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

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

#### **Multi** Recherche dans les sacs de données

Vous pouvez vous renseigner sur:

- Valeur d'un élément entier à l'aide de l'appel `mqInquireInteger`. Voir [mqInquireInteger](#).
- Valeur d'un élément entier 64 bits à l'aide de l'appel `mqInquireInteger64`. Voir [mqInquireInteger64](#).
- Valeur d'un élément de filtre d'entier à l'aide de l'appel `mqInquireIntegerFilter`. Voir [mqInquireIntegerFilter](#).
- Valeur d'un élément de chaîne de caractères à l'aide de l'appel de chaîne `mqInquire`. Voir [mqInquireString](#).

- Valeur d'un élément de filtre de chaîne à l'aide de l'appel `mqInquireStringFilter`. Voir [mqInquireStringFilter](#).
- Valeur d'un élément de chaîne d'octets à l'aide de l'appel `mqInquireByteString`. Voir [mqInquireByteString](#).
- Valeur d'un élément de filtre de chaîne d'octet à l'aide de l'appel de filtre `mqInquireByteStringFilter`. Voir [mqInquireByteStringFilter](#).
- Valeur d'un descripteur de sac à l'aide de l'appel de sac `mqInquire`. Voir [mqInquireBag](#).

Vous pouvez également vous renseigner sur le type (entier, entier 64 bits, filtre entier, chaîne de caractères, filtre de chaîne, chaîne d'octets, filtre de chaîne d'octets ou descripteur de sac) d'un élément spécifique à l'aide de l'appel `mqInquireItemInfo`. Voir [mqInquireItemInfo](#).

### Multi Modification d'informations dans un sac

L'interface MQAI vous permet de modifier des informations dans un sac à l'aide des appels `mqSet*`. Vous pouvez :

1. Modifier des éléments de données dans un sac. L'index permet de remplacer une instance individuelle d'un paramètre en identifiant l'occurrence de l'élément à modifier (voir [Figure 6](#), à la page 72).

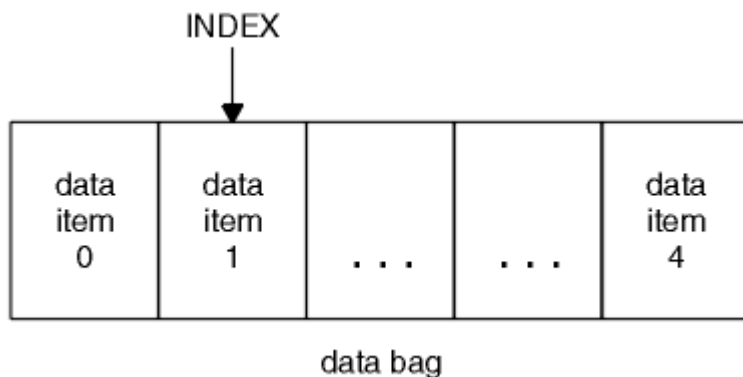


Figure 6. Modification d'un élément de données unique

2. Supprimez toutes les occurrences existantes du sélecteur spécifié et ajoutez une nouvelle occurrence à la fin du sac. (Voir [Figure 7](#), à la page 72.) Une valeur d'index spéciale permet de remplacer **toutes les instances** d'un paramètre.

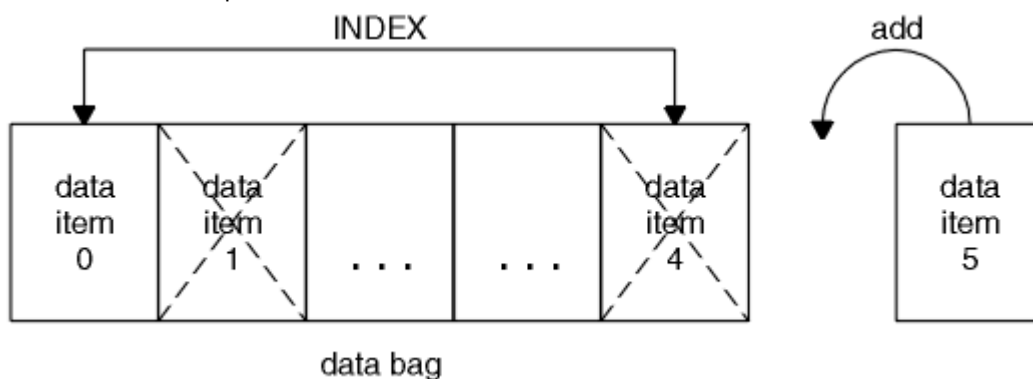


Figure 7. Modification de tous les éléments de données

**Remarque :** L'index conserve l'ordre d'insertion dans le sac mais peut affecter les index d'autres éléments de données.

L'appel `mqSetInteger` vous permet de modifier des éléments de type entier dans un sac. L'appel `mqSetInteger64` vous permet de modifier des éléments d'entier 64 bits. L'appel `mqSetIntegerFilter` vous permet de modifier des éléments de filtre de type entier. L'appel de chaîne `mqSet` vous permet de modifier des éléments de chaîne de caractères. L'appel `mqSetStringFilter` vous permet de modifier des éléments de filtre de chaîne. L'appel `mqSetByteString` vous permet de modifier des éléments de chaîne d'octets.



L'appel de filtre `mqSetByteStringvous` permet de modifier des éléments de filtre de chaîne d'octets. Vous pouvez également utiliser ces appels pour supprimer toutes les occurrences existantes du sélecteur spécifié et ajouter une nouvelle occurrence à la fin du sac. L'élément de données peut être un élément utilisateur ou un élément système.

Pour une description complète de ces appels, voir:

- [mqSetentier](#)
- [mqSetInteger64](#)
- [mqSetIntegerFilter](#)
- [mqSetChaîne](#)
- [mqSetStringFilter](#)
- [mqSetByteString](#)
- [mqSetByteStringFiltre](#)

**Multi** *Effacement d'un sac à l'aide de l'appel de sac `mqClear`*

L'appel de sac `mqClear` supprime tous les éléments utilisateur d'un sac utilisateur et réinitialise les éléments système à leurs valeurs initiales. Les sacs système contenus dans le sac sont également supprimés.

Pour une description complète de l'appel de sac `mqClear`, voir [mqClearBag](#).

**Multi** *Troncation d'un sac à l'aide de l'appel de sac `mqTruncate`*

L'appel de sac `mqTruncate` réduit le nombre d'éléments utilisateur dans un sac utilisateur en supprimant les éléments de la fin du sac, en commençant par l'élément ajouté le plus récemment. Par exemple, il peut être utilisé lors de l'utilisation des mêmes informations d'en-tête pour générer plusieurs messages.

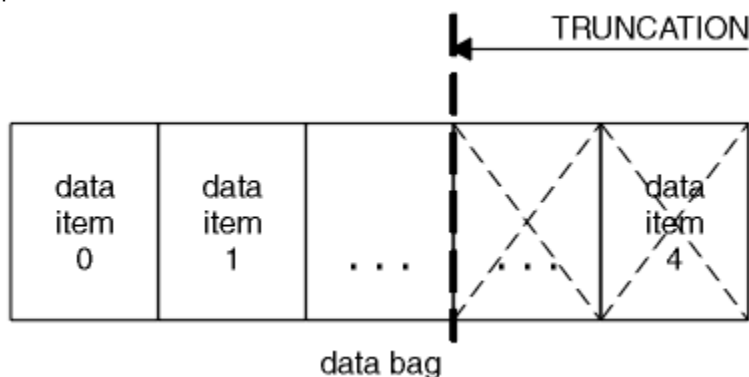


Figure 8. Troncation d'un sac

Pour une description complète de l'appel de sac `mqTruncate`, voir [mqTruncateBag](#).

**Multi** *Conversion de sacs et de tampons*

Pour envoyer des données entre applications, on place d'abord les données de message dans un sac. Ensuite, les données du sac sont converties en message PCF à l'aide de l'appel `mqBagToBuffer`. Le message PCF est envoyé à la file d'attente requise à l'aide de l'appel `MQPUT`. Cela est illustré dans la figure [Figure 9](#), à la page 74. Pour une description complète de l'appel `mqBagToBuffer`, voir [mqBagToBuffer](#).

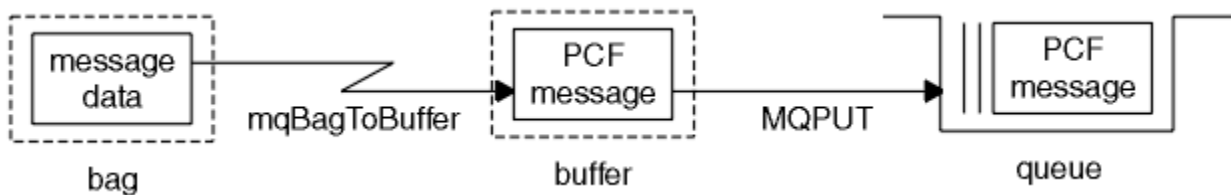


Figure 9. Conversion de sacs en messages PCF

Pour recevoir des données, le message est reçu dans une mémoire tampon à l'aide de l'appel MQGET. Les données de la mémoire tampon sont ensuite converties en un sac à l'aide de l'appel `mqBufferToBag`, à condition que la mémoire tampon contienne un message PCF valide. Cela est illustré dans la figure Figure 10, à la page 74. Pour une description complète de l'appel `mqBufferToBag`, voir [mqBufferToBag](#).

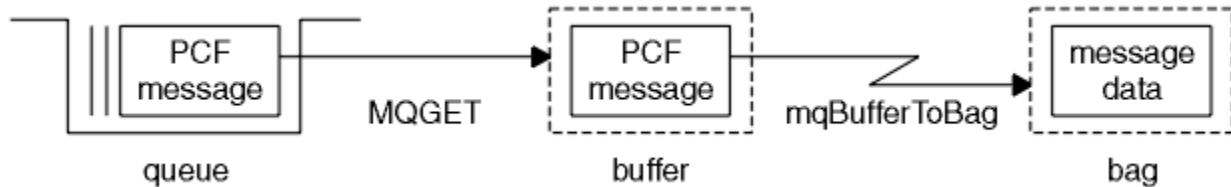


Figure 10. Conversion de messages PCF en forme de sac

#### Multi Comptage des éléments de données

L'appel d'éléments `mqCount` compte le nombre d'éléments utilisateur, d'éléments système, ou les deux, qui sont stockés dans un sac de données et renvoie ce nombre. Par exemple, `mqCountItems ( Bag , 7 , ... )` renvoie le nombre d'éléments dans le sac avec un sélecteur de 7. Il peut compter des éléments par sélecteur individuel, par sélecteur d'utilisateur, par sélecteur système ou par tous les sélecteurs.

**Remarque :** Cet appel compte le nombre d'éléments de données, et non le nombre de sélecteurs uniques dans le sac. Un sélecteur peut se produire plusieurs fois, de sorte qu'il peut y avoir moins de sélecteurs uniques dans le sac que d'éléments de données.

Pour une description complète de l'appel `mqCountItems`, voir [mqCountItems](#).

#### Multi Suppression d'éléments de données

Vous pouvez supprimer des éléments de sacs de plusieurs manières. Vous pouvez :

- Retirez un ou plusieurs éléments utilisateur d'un sac. Pour des informations détaillées, voir «Suppression d'éléments de données d'un sac à l'aide de l'appel d'élément `mqDelete`», à la page 74.
- Supprimez tous les éléments utilisateur d'un sac, c'est-à-dire, effacez un sac. Pour plus d'informations, voir «Effacement d'un sac à l'aide de l'appel de sac `mqClear`», à la page 73.
- Supprimer des éléments utilisateur de la fin d'un sac, c'est-à-dire tronquer un sac. Pour des informations détaillées, voir «Troncation d'un sac à l'aide de l'appel de sac `mqTruncate`», à la page 73.

#### Multi Suppression d'éléments de données d'un sac à l'aide de l'appel d'élément `mqDelete`

L'appel d'élément `mqDelete` supprime un ou plusieurs éléments utilisateur d'un sac. L'index est utilisé pour supprimer:

1. Une seule occurrence du sélecteur spécifié. (Voir Figure 11, à la page 75.)

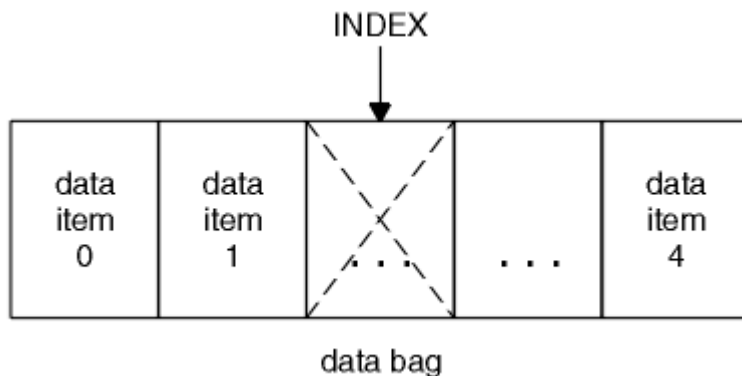


Figure 11. Suppression d'un élément de données unique

ou

2. Toutes les occurrences du sélecteur spécifié. (Voir Figure 12, à la page 75.)

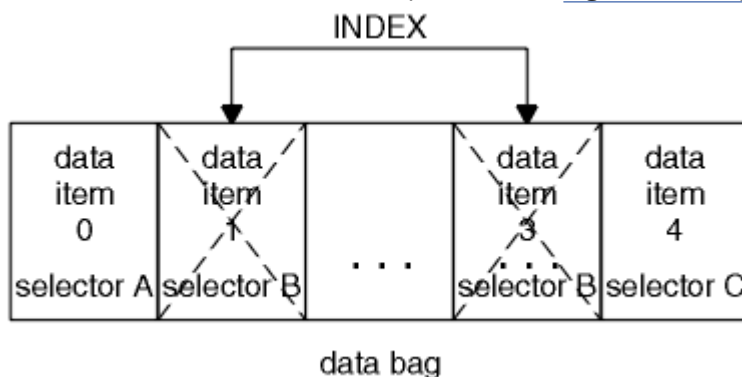


Figure 12. Suppression de tous les éléments de données

**Remarque :** L'index conserve l'ordre d'insertion dans le sac mais peut affecter les index d'autres éléments de données. Par exemple, l'appel d'élément `mqDelete` ne conserve pas les valeurs d'index des éléments de données qui suivent l'élément supprimé car les index sont réorganisés pour combler l'écart qui reste de l'élément supprimé.

Pour une description complète de l'appel d'élément `mqDelete`, voir [mqDeleteItem](#).

## **Multi** Envoi de commandes d'administration au serveur de commandes `qm` à l'aide de l'appel `mqExecute`

Lorsqu'un sac de données a été créé et rempli, un message de commande d'administration peut être envoyé au serveur de commandes d'un gestionnaire de files d'attente à l'aide de l'appel `mqExecute`. Cela permet de gérer l'échange avec le serveur de commandes et de renvoyer les réponses dans un sac.

Une fois que vous avez créé et rempli votre sac de données, vous pouvez envoyer un message de commande d'administration au serveur de commandes d'un gestionnaire de files d'attente. La méthode la plus simple consiste à utiliser l'appel `mqExecute`. L'appel `mqExecute` envoie un message de commande d'administration en tant que message non persistant et attend les réponses. Les réponses sont renvoyées dans un sac de réponse. Ils peuvent contenir des informations sur les attributs relatifs à plusieurs objets IBM MQ ou à une série de messages de réponse d'erreur PCF, par exemple. Par conséquent, le sac de réponse peut contenir un code retour uniquement ou *des sacs imbriqués*.

Les messages de réponse sont placés dans des sacs système créés par le système. Par exemple, pour les demandes concernant les noms d'objets, un sac système est créé pour contenir ces noms d'objet et le sac est inséré dans le sac utilisateur. Les poignées de ces sacs sont ensuite insérées dans le sac de réponse et le sac imbriqué est accessible par le sélecteur `MQHA_BAG_HANDLE`. Le sac système reste en stockage, s'il n'est pas supprimé, jusqu'à ce que le sac de réponse soit supprimé.

Le concept d' *imbrication* est illustré dans la [Figure 13](#), à la page 76.

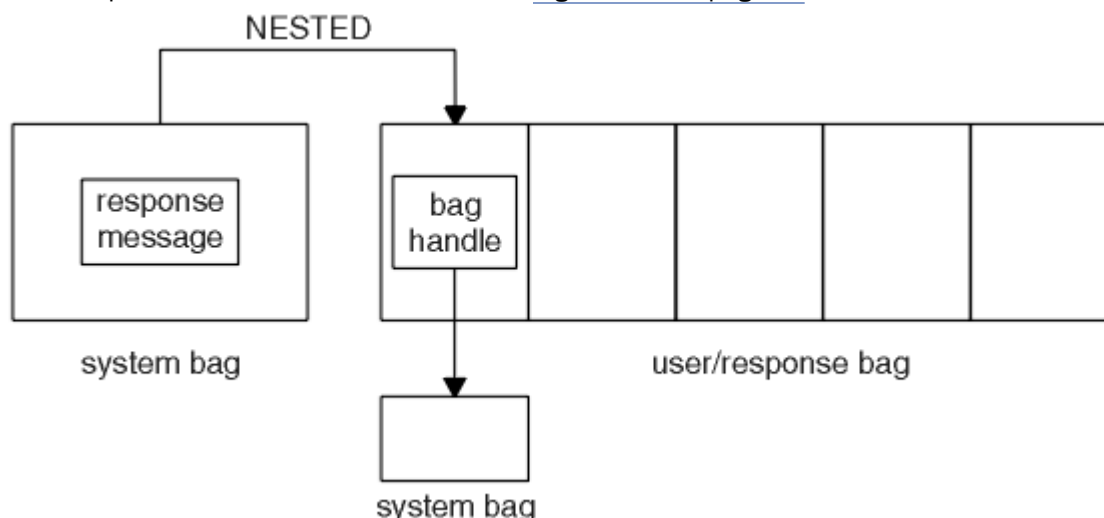


Figure 13. Imbrication

En tant qu'entrée de l'appel `mqExecute`, vous devez fournir:

- Un descripteur de connexion MQI.
- Commande à exécuter. Il doit s'agir de l'une des valeurs `MQCMD_*`.

**Remarque :** Si cette valeur n'est pas reconnue par MQAI, elle est toujours acceptée. Toutefois, si l'appel d'interrogation `mqAdda` été utilisé pour insérer des valeurs dans le sac, ce paramètre doit être une commande `INQUIRE` reconnue par MQAI. Autrement dit, le paramètre doit être de la forme `MQCMD_INQUIRE_*`.

- Eventuellement, une poignée du sac contenant des options qui contrôlent le traitement de l'appel. C'est également là que vous pouvez spécifier la durée maximale en millisecondes pendant laquelle MQAI doit attendre chaque message de réponse.
- Descripteur du sac d'administration qui contient les détails de la commande d'administration à émettre.
- Descripteur du sac de réponse qui reçoit les messages de réponse.

Les descripteurs suivants sont facultatifs:

- Descripteur d'objet de la file d'attente dans laquelle la commande d'administration doit être placée.  
Si aucun descripteur d'objet n'est spécifié, la commande d'administration est placée sur `SYSTEM.ADMIN.COMMAND.QUEUE` appartenant au gestionnaire de files d'attente actuellement connecté. Il s'agit de l'option par défaut.
- Descripteur d'objet de la file d'attente dans laquelle les messages de réponse doivent être placés.


Vous pouvez choisir de placer les messages de réponse dans une file d'attente dynamique créée automatiquement par MQAI. La file d'attente créée existe uniquement pour la durée de l'appel et est supprimée par l'interface MQAI à la sortie de l'appel `mqExecute`.

Pour obtenir des exemples d'utilisation de l'appel `mqExecute`, voir [Exemple de code](#)

## Administration à l'aide de REST API

Vous pouvez utiliser l'administrative REST API pour administrer les objets IBM MQ, tels que les gestionnaires de files d'attente et les files d'attente, ainsi que les agents et les transferts Managed File Transfer. Les informations sont envoyées et reçues du administrative REST API au format JSON. Ces API RESTful peuvent vous aider à intégrer l'administration IBM MQ dans des DevOps et des outils d'automatisation populaires.

## Avant de commencer

**Remarque :**  administrative REST API n'est pas disponible dans une installation IBM MQ Web Server autonome. Pour plus d'informations sur les options d'installation du composant IBM MQ qui exécute le administrative REST API, voir [IBM MQ Console et REST API](#).

Pour des informations de référence sur les ressources REST disponibles, voir [The administrative REST API reference](#).

## Procédure




- «[Guide d'initiation à administrative REST API](#)», à la page 77
- «[Utilisation de administrative REST API](#)», à la page 80
- «[Administration à distance à l'aide de REST API](#)», à la page 82
- «[REST API Horodatages](#)», à la page 87
- «[REST API traitement des erreurs](#)», à la page 87
- «[REST API découverte](#)», à la page 90
- «[REST APISupport de langue nationale \(NLS\)](#)», à la page 91

## Guide d'initiation à administrative REST API

Démarrez rapidement avec administrative REST API et essayez quelques exemples de demande à l'aide de cURL pour créer, mettre à jour, afficher et supprimer une file d'attente.

### Avant de commencer


Pour vous initier à l'utilisation du administrative REST API, les exemples de cette tâche comportent les exigences suivantes:

- Les exemples utilisent cURL pour effectuer des demandes REST afin d'afficher des informations sur les gestionnaires de files d'attente sur le système et pour créer une file d'attente, mettre à jour, afficher et supprimer une file d'attente. Par conséquent, pour effectuer cette tâche, vous devez installer cURL sur votre système.
- Pour effectuer cette tâche, vous devez être un utilisateur doté de certains privilèges vous permettant d'employer la commande **dspmqweb** :
  -  Sur z/OS, vous devez disposer des droits permettant d'exécuter la commande **dspmqweb** et de l'accès en écriture sur le fichier `mqwebuser.xml`.
  -  Sur tous les autres systèmes d'exploitation, vous devez être un [utilisateur privilégié](#).
  -  Sous IBM i, les commandes doivent être exécutées dans QSHELL.

## Procédure

1. Vérifiez que vous avez configuré le serveur mqweb pour qu'il soit utilisé par administrative REST API, administrative REST API for MFT, messaging REST API ou IBM MQ Console.

Pour plus d'informations sur la configuration du serveur mqweb avec un registre de base, voir [Configuration de base pour le serveur mqweb](#).

2.  Sur z/OS, définissez la variable d'environnement WLP\_USER\_DIR afin de pouvoir utiliser la commande **dspmqweb**. Définissez la variable de telle sorte qu'elle pointe vers votre configuration de serveur mqweb à l'aide de la commande suivante :

```
export WLP_USER_DIR=WLP_user_directory
```

où `WLP_user_directory` est le nom du répertoire transmis à `crtmqweb`. Par exemple :

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

Pour plus d'informations, voir [Creating the mqweb server](#).

3. Déterminez l'URL REST API en entrant la commande suivante:

```
dspmweb status
```

Les exemples des étapes suivantes supposent que votre URL REST API est l'URL par défaut `https://localhost:9443/ibmmq/rest/v1/`. Si votre URL est différente de l'URL par défaut, remplacez-la dans les étapes ci-après.

4. Essayez une demande GET sur la ressource `qmgr` en utilisant l'authentification de base avec l'utilisateur `mqadmin` :

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

5. Créez, affichez, modifiez et supprimez une file d'attente à l'aide de la ressource `mqsc` :

Cet exemple utilise un gestionnaire de files d'attente QM1. Créez un gestionnaire de files d'attente portant le même nom ou remplacez un gestionnaire de files d'attente existant sur votre système.

- a) Effectuez une demande POST sur la ressource `mqsc` pour créer la file d'attente locale:

Dans le corps de la demande, le nom de la nouvelle file d'attente est défini sur `Q1`. L'authentification de base est utilisée et un en-tête HTTP `ibm-mq-rest-csrf-token` avec une valeur arbitraire est défini dans la demande REST cURL . Cet en-tête supplémentaire est requis pour les demandes POST, PATCH et DELETE:

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

- b) Effectuez une demande POST sur la ressource `mqsc` pour afficher la file d'attente locale créée à l'étape «5.a», à la page 78:

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

- c) Effectuez une demande POST sur la ressource `mqsc` vers la ressource pour mettre à jour la description de la file d'attente:

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

- d) Effectuez une demande POST sur la ressource `mqsc` pour afficher la nouvelle description de file d'attente. Spécifiez l'attribut **responseParameters** dans le corps de la demande de sorte que la réponse contienne la zone de description:

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

- e) Effectuez une demande POST sur la ressource `mqsc` pour supprimer la file d'attente:

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

- f) Effectuez une demande POST sur la ressource mqsc pour prouver que la file d'attente a été supprimée:

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



## Que faire ensuite

- Les exemples utilisent l'authentification de base pour sécuriser la demande. Vous pouvez utiliser à la place l'authentification basée sur un jeton ou l'authentification basée sur le client. Pour plus d'informations, voir [Utilisation de l'authentification par certificat client avec l'REST API et IBM MQ Console](#) et [Utilisation de l'authentification basée sur un jeton avec l'REST API](#).
- En savoir plus sur l'utilisation de l'administrative REST API et la construction d'URL avec des paramètres de requête: «[Utilisation de l'administrative REST API](#)», à la page 80.
- Parcourez les informations de référence pour les ressources administrative REST API disponibles et tous les paramètres de requête facultatifs disponibles: [administrative REST API reference](#).
- Apprenez à utiliser l'administrative REST API pour administrer des objets IBM MQ sur des systèmes distants: «[Administration à distance à l'aide de REST API](#)», à la page 82.
- Apprenez à utiliser l'administrative REST API avec MFT:«[Initiation à REST API for MFT](#)», à la page 79.
- Découvrez messaging REST API, une interface RESTful pour la messagerie IBM MQ : [Messaging à l'aide de REST API](#).
- Découvrez IBM MQ Console, une interface graphique basée sur un navigateur: «[Administration à l'aide de IBM MQ Console](#)», à la page 94.

## Initiation à REST API for MFT

Initiez-vous rapidement à l'administrative REST API for Managed File Transfer et essayez quelques exemples de demandes pour afficher le statut de l'agent MFT et pour afficher la liste des transferts.

## Avant de commencer

- Les exemples utilisent cURL pour envoyer des demandes REST afin d'afficher la liste des transferts et le statut de l'agent MFT . Par conséquent, pour effectuer cette tâche, vous devez installer cURL sur votre système.
  - Pour effectuer cette tâche, vous devez être un utilisateur doté de certains privilèges vous permettant d'employer la commande **dspmweb** :
-  Sur z/OS, vous devez disposer des droits permettant d'exécuter la commande **dspmweb** et de l'accès en écriture sur le fichier `mqwebuser.xml`.
  -  Sur tous les autres systèmes d'exploitation, vous devez être un [utilisateur privilégié](#).

## Procédure

1. Vérifiez que le serveur mqweb est configuré pour l'administrative REST API for MFT:

- Vérifiez que vous avez configuré le serveur mqweb pour qu'il soit utilisé par l'administrative REST API, l'administrative REST API for MFT, le messaging REST API ou IBM MQ Console. Pour plus d'informations sur la configuration du serveur mqweb avec un registre de base, voir [Configuration de base pour le serveur mqweb](#).
- Si le serveur mqweb est configuré, vérifiez que l'étape 8 de la [configuration de base du serveur mqweb](#) a été effectuée pour activer l'administrative REST API for MFT.

2. 

Sur z/OS, définissez la variable d'environnement `WLP_USER_DIR` afin de pouvoir utiliser la commande **dspmweb**. Définissez la variable de telle sorte qu'elle pointe vers votre configuration de serveur mqweb à l'aide de la commande suivante :

```
export WLP_USER_DIR=WLP_user_directory
```

où `WLP_user_directory` est le nom du répertoire transmis à `crtmqweb`. Par exemple :

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

Pour plus d'informations, voir [Creating the mqweb server](#).

3. Déterminez l'URL REST API en entrant la commande suivante:

```
dspmweb status
```

Les exemples des étapes suivantes supposent que votre URL REST API est l'URL par défaut `https://localhost:9443/ibmmq/rest/v1/`. Si votre URL est différente de l'URL par défaut, remplacez-la dans les étapes ci-après.

4. Effectuez une demande GET sur la ressource `agent` pour renvoyer des détails de base sur tous les agents, y compris le nom, le type et l'état:

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

5. Créez des transferts à afficher à l'aide de la commande **fteCreateTransfer**.

Le serveur mqweb met en cache des informations sur les transferts et renvoie ces informations lorsqu'une demande est effectuée. Ce cache est réinitialisé lorsque le serveur mqweb est redémarré. Vous pouvez voir si le serveur a été redémarré en affichant les fichiers `console.log` et messages `logou` sous z/OS, en examinant la sortie de la tâche démarrée.

6. Effectuez une demande GET sur la ressource `transfer` pour renvoyer les détails d'un maximum de quatre transferts effectués depuis le démarrage du serveur mqweb:

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

## Que faire ensuite

- Les exemples utilisent l'authentification de base pour sécuriser la demande. Vous pouvez utiliser à la place l'authentification basée sur un jeton ou l'authentification basée sur le client. Pour plus d'informations, voir [Utilisation de l'authentification basée sur un jeton avec l'REST API](#) et [Utilisation de l'authentification par certificat client avec l'REST API et IBM MQ Console](#).
- En savoir plus sur l'utilisation de l'administrative REST API et la construction d'URL avec des paramètres de requête: [«Utilisation de l'administrative REST API», à la page 80](#).
- Parcourez les informations de référence des ressources administrative REST API for MFT disponibles et tous les paramètres de requête facultatifs disponibles: [administrative REST API reference](#).
- Découvrez messaging REST API, une interface RESTful pour la messagerie IBM MQ : [Messaging à l'aide de REST API](#).
- Découvrez IBM MQ Console, une interface graphique basée sur un navigateur: [«Administration à l'aide de IBM MQ Console», à la page 94](#).

## Utilisation de l'administrative REST API

Lorsque vous utilisez l'administrative REST API, vous appelez des méthodes HTTP sur des URL qui représentent les différents objets IBM MQ, tels que les gestionnaires de files d'attente ou les files d'attente. La méthode HTTP, par exemple POST, représente le type d'action à effectuer sur l'objet représenté par l'URL. Des informations supplémentaires sur l'action peuvent être fournies en JSON dans



le cadre du contenu de la méthode HTTP ou codées dans des paramètres de requête. Des informations sur le résultat de l'exécution de l'action peuvent être renvoyées en tant que corps de la réponse HTTP.

## Avant de commencer

Tenez compte des points suivants avant d'utiliser administrative REST API:

- Vous devez vous authentifier auprès du serveur mqweb pour pouvoir utiliser administrative REST API. Vous pouvez vous authentifier à l'aide de l'authentification de base HTTP, de l'authentification par certificat client ou de l'authentification par jeton. Pour plus d'informations sur l'utilisation de ces méthodes d'authentification, voir [Sécurité IBM MQ Console et REST API](#).
- REST API est sensible à la casse. Par exemple, une requête HTTP GET sur l'URL suivante n'affiche pas d'informations si le gestionnaire de files d'attente est appelé `qmgr1`.

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

- Tous les caractères pouvant être utilisés dans les noms d'objet IBM MQ ne peuvent pas être directement codés dans une URL. Pour coder correctement ces caractères, vous devez utiliser le codage d'URL approprié:
  - Une barre oblique (/) doit être codée sous la forme %2F.
  - Le signe de pourcentage, %, doit être codé en tant que %25.
- En raison du comportement de certains navigateurs, ne nommez pas les objets en utilisant uniquement des points ou des barres obliques.

## Pourquoi et quand exécuter cette tâche

Lorsque vous utilisez REST API pour effectuer une action sur un objet, vous devez d'abord construire une URL pour représenter cet objet. Chaque URL commence par un préfixe, qui décrit le nom d'hôte et le port auxquels envoyer la demande. Le reste de l'URL décrit un objet particulier, ou un ensemble d'objets, appelé ressource.

L'action à effectuer sur la ressource définit si l'URL a besoin ou non de paramètres de requête. Il définit également la méthode HTTP utilisée et indique si des informations supplémentaires sont envoyées à l'URL, ou renvoyées à partir de celle-ci, au format JSON. Les informations supplémentaires peuvent faire partie de la demande HTTP ou être renvoyées dans le cadre de la réponse HTTP.

Après avoir construit l'URL et créé un contenu JSON facultatif pour l'envoi dans la demande HTTP, vous pouvez envoyer la demande HTTP à IBM MQ. Vous pouvez envoyer la demande à l'aide de l'implémentation HTTP qui est intégrée au langage de programmation de votre choix. Vous pouvez également envoyer les demandes à l'aide d'outils de ligne de commande tels que cURL, d'un navigateur Web ou d'un module complémentaire de navigateur Web.

**Important :** Vous devez au minimum effectuer les étapes [«1.a»](#), à la page 81 et [«1.b»](#), à la page 81.

## Procédure

1. Construisez l'URL:

- a) Déterminez l'URL de préfixe en entrant la commande suivante:

```
dspmweb status
```

L'URL que vous souhaitez utiliser inclut la phrase `/ibmmq/rest/`.

- b) Ajoutez la ressource au chemin d'URL.

Les ressources IBM MQ suivantes sont disponibles:

- [/admin/installation](#)
- [/admin/gestionnaire\\_files](#)
- [/admin/file d'attente](#)

- [/admin/abonnement](#)
- [/admin/canal](#)
- [/action/qmgr/{qmgrname}/mqsc](#)

Les ressources Managed File Transfer suivantes sont disponibles:

- [/admin/agent](#)
- [/admin/transfert](#)
- [/admin/moniteur](#)

Par exemple, pour interagir avec les gestionnaires de files d'attente, ajoutez /qmgr à l'URL de préfixe pour créer l'URL suivante:

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

c) Facultatif : Ajoutez des segments de chemin facultatifs supplémentaires à l'URL.

Dans les informations de référence de chaque type d'objet, les segments facultatifs peuvent être identifiés dans l'URL par les accolades qui l'entourent { }.

Par exemple, ajoutez le nom de gestionnaire de files d'attente QM1 à l'URL pour créer l'URL suivante:

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

d) Facultatif : Ajoutez un paramètre de requête facultatif à l'URL.

Ajouter un point d'interrogation,?, nom de variable, signe égal =, et une valeur ou une liste de valeurs à l'URL.

Par exemple, pour demander tous les attributs du gestionnaire de files d'attente QM1, créez l'URL suivante:

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

e) Ajoutez d'autres paramètres de requête facultatifs à l'URL.

Ajoutez une perluète, &, à l'URL, puis répétez l'étape d.

2. Appelez la méthode HTTP appropriée sur l'URL. Spécifiez un contenu JSON facultatif et fournissez les données d'identification de sécurité appropriées pour l'authentification. Exemple :

- Utilisez l'implémentation HTTP/REST du langage de programmation de votre choix.
- Utilisez un outil tel qu'un module complémentaire de navigateur client REST ou cURL.

## Administration à distance à l'aide de REST API

Vous pouvez utiliser REST API pour administrer les gestionnaires de files d'attente éloignées et les objets IBM MQ qui sont associés à ces gestionnaires de files d'attente. Cette administration à distance inclut les gestionnaires de files d'attente qui se trouvent sur le même système, mais pas dans la même installation IBM MQ que le serveur mqweb. Par conséquent, vous pouvez utiliser l'REST API pour administrer l'intégralité de votre réseau IBM MQ avec une seule installation qui exécute le serveur mqweb. Pour administrer les gestionnaires de files d'attente éloignées, vous devez configurer la passerelle administrative REST API de sorte qu'au moins un gestionnaire de files d'attente de la même installation que le serveur mqweb fasse office de gestionnaire de files d'attente de passerelle. Vous pouvez ensuite spécifier le gestionnaire de files d'attente éloignées dans l'URL de ressource REST API pour effectuer l'action d'administration spécifiée.

### Avant de commencer

Vous pouvez empêcher l'administration à distance en désactivant la passerelle administrative REST API . Pour plus d'informations, voir [Configuration de la passerelle administrative REST API](#).

Pour utiliser la passerelle administrative REST API , les conditions suivantes doivent être remplies:

- Le serveur mqweb doit être configuré et démarré. Pour plus d'informations sur la configuration et le démarrage du serveur mqweb, voir [«Guide d'initiation à administrative REST API»](#), à la page 77.
- Le gestionnaire de files d'attente que vous souhaitez configurer en tant que gestionnaire de files d'attente de passerelle doit se trouver dans la même installation que le serveur mqweb.
- Le gestionnaire de files d'attente éloignées que vous souhaitez administrer doit être IBM MQ 8.0 ou une version ultérieure.
- Vous devez vous assurer que tous les attributs spécifiés dans votre demande sont valides pour le système auquel vous envoyez la demande. Par exemple, si le gestionnaire de files d'attente de passerelle est sous Windows et que le gestionnaire de files d'attente éloignées est sous z/OS, vous ne pouvez pas demander que l'attribut `dataCollection.statistics` soit renvoyé pour une demande HTTP GET sur la ressource `queue`.
- Vous devez vous assurer que tous les attributs spécifiés dans votre demande sont valides pour le niveau de IBM MQ auquel vous envoyez la demande. Par exemple, si le gestionnaire de files d'attente éloignées exécute IBM MQ 8.0, vous ne pouvez pas demander que l'attribut `extended.enableMediaImageOperations` soit renvoyé pour une demande HTTP GET sur la ressource `queue`.
- Vous devez utiliser l'une des ressources REST prises en charge suivantes:
  - `/queue`
  - `/subscription`
  - `/channel`
  - `/mqsc`
  - `/qmgr`

La ressource `/qmgr` renvoie uniquement un sous-ensemble des attributs lorsque vous interrogez un gestionnaire de files d'attente éloignées: `name`, `status.started`, `status.channelInitiatorState`, `status.ldapConnectionState`, `status.connectionCount` et `status.publishSubscribeState`.

## Pourquoi et quand exécuter cette tâche

Pour utiliser la passerelle administrative REST API afin d'administrer les gestionnaires de files d'attente éloignées, vous devez préparer les gestionnaires de files d'attente pour l'administration à distance. Autrement dit, vous devez configurer les files d'attente de transmission, les programmes d'écoute et les canaux émetteur et récepteur entre le gestionnaire de files d'attente de passerelle et le gestionnaire de files d'attente éloignées. Vous pouvez ensuite envoyer une demande REST au gestionnaire de files d'attente éloignées en spécifiant le gestionnaire de files d'attente dans l'URL de la ressource. Le gestionnaire de files d'attente de passerelle est spécifié à l'aide de la commande **setmqweb** pour définir l'attribut `mqRestGatewayQmgr` sur le nom du gestionnaire de files d'attente de passerelle ou en envoyant le nom du gestionnaire de files d'attente de passerelle dans un en-tête envoyé avec la demande. La demande est envoyée via le gestionnaire de files d'attente de passerelle au gestionnaire de files d'attente éloignées. La réponse est renvoyée avec un en-tête indiquant le gestionnaire de files d'attente qui a été utilisé comme gestionnaire de files d'attente de passerelle.

## Procédure

1. Configurez les communications entre le gestionnaire de files d'attente de passerelle et les gestionnaires de files d'attente éloignées que vous souhaitez administrer. Ces étapes de configuration sont les mêmes que celles requises pour configurer l'administration à distance par `runmqsc` et PCF. Pour plus d'informations sur ces étapes, voir [«Configuration des gestionnaires de files d'attente pour l'administration à distance»](#), à la page 207.
2. Configurez la sécurité sur les gestionnaires de files d'attente éloignées:

- a) Vérifiez que les ID utilisateur appropriés existent sur le système sur lequel s'exécute le gestionnaire de files d'attente éloignées. L'ID utilisateur qui doit exister sur le système distant dépend du rôle de l'utilisateur REST API :
- Si l'utilisateur REST API se trouve dans le groupe MQWebAdmin ou MQWebAdminRO, l'ID utilisateur qui a démarré le serveur mqweb doit exister sur le système distant. Sur le IBM MQ Appliance, l'utilisateur qui démarre le serveur mqweb est mqsystem.
  - Si l'utilisateur REST API appartient au groupe MQWebUser, cet ID utilisateur REST API doit exister sur le système distant.
- b) Assurez-vous que les ID utilisateur appropriés disposent des niveaux d'autorisation nécessaires pour accéder aux ressources REST API appropriées sur le gestionnaire de files d'attente éloignées:
- Droit d'insertion de messages dans SYSTEM.ADMIN.COMMAND.QUEUE.
  - Droit d'insertion de messages dans SYSTEM.REST.REPLY.QUEUE.
  - Droit d'accès aux files d'attente de transmission définies pour l'administration à distance.
  - Droit d'affichage des attributs du gestionnaire de files d'attente.
  - Droit d'exécution des demandes REST. Pour plus d'informations, voir la section Security requirements des rubriques de référence sur les ressources REST API.
3. Configurez le gestionnaire de files d'attente local utilisé comme passerelle. Vous pouvez configurer un gestionnaire de files d'attente de passerelle par défaut, spécifier le gestionnaire de files d'attente de passerelle dans un en-tête HTTP ou utiliser une combinaison des deux approches:
- Configurez un gestionnaire de files d'attente de passerelle par défaut à l'aide de la commande **setmqweb** :

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

où *qmgrName* est le nom du gestionnaire de files d'attente de passerelle.

Ce gestionnaire de files d'attente de passerelle est utilisé lorsque les deux instructions suivantes sont vraies:

- Un gestionnaire de files d'attente n'est pas spécifié dans l'en-tête `ibm-mq-rest-gateway-qmgr` d'une demande REST.
  - Le gestionnaire de files d'attente spécifié dans l'URL de ressource REST API n'est pas un gestionnaire de files d'attente local.
- Configurez le gestionnaire de files d'attente de passerelle sur chaque demande REST en définissant l'en-tête HTTP `ibm-mq-rest-gateway-qmgr` sur le nom du gestionnaire de files d'attente de passerelle.
4. Incluez le nom du gestionnaire de files d'attente éloignées que vous souhaitez administrer dans l'URL de la ressource.

Par exemple, pour obtenir la liste des files d'attente du gestionnaire de files d'attente éloignées `remoteQM`, utilisez l'URL suivante:

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

## Résultats

Un en-tête `ibm-mq-rest-gateway-qmgr` est renvoyé avec la réponse REST. Cet en-tête indique le gestionnaire de files d'attente qui a été utilisé comme gestionnaire de files d'attente de passerelle.

Si vous avez des difficultés à utiliser administrative REST API pour administrer les gestionnaires de files d'attente éloignées:

- Vérifiez que le gestionnaire de files d'attente éloignées est en cours d'exécution.
- Vérifiez que le serveur de commandes est en cours d'exécution sur le système distant.

- Vérifiez que l'intervalle de déconnexion du canal n'a pas expiré. Par exemple, si un canal a démarré, puis s'est arrêté après un certain temps. Ceci est particulièrement important si vous démarrez les canaux manuellement.

### Exemple

Dans l'exemple suivant, il existe trois installations IBM MQ sur deux machines. Sous Machine 1, il existe un Installation 1 et un Installation 2. Sous Machine 2, il existe un Installation 3. Un serveur mqweb est configuré pour Installation 1. Il existe un seul gestionnaire de files d'attente dans chaque installation et ces gestionnaires de files d'attente sont configurés pour l'administration à distance. C'est-à-dire que les programmes d'écoute, les canaux et les files d'attente suivants sont configurés et démarrés:

- Sur le gestionnaire de files d'attente QM1, dans Installation 1, sous Machine 1:
  - Canal émetteur QM1.to.QM2
  - Canal récepteur QM2.to.QM1
  - Canal émetteur QM1.to.QM3
  - Canal récepteur QM3.to.QM1
  - File d'attente de transmission QM2
  - File d'attente de transmission QM3
  - Un programme d'écoute configuré sur le port 1414
- Sur le gestionnaire de files d'attente QM2, dans Installation 2, sous Machine 1:
  - Canal émetteur QM2.to.QM1.
  - Canal récepteur QM1.to.QM2
  - File d'attente de transmission QM1
  - Un programme d'écoute configuré sur le port 1415
- Sur le gestionnaire de files d'attente QM3, dans Installation 3, sous Machine 2:
  - Canal émetteur QM3.to.QM1
  - Canal récepteur QM1.to.QM3
  - File d'attente de transmission QM1
  - Le programme d'écoute par défaut

Une file d'attente, Qon2 , est définie sur QM2et une file d'attente Qon3 est définie sur QM3.

L'utilisateur mquser est défini sur les deux machines, reçoit le rôle MQWebAdmin dans REST APIet est autorisé à accéder aux files d'attente appropriées sur chaque gestionnaire de files d'attente.

La commande setmqweb permet de configurer le gestionnaire de files d'attente QM1 comme gestionnaire de files d'attente de passerelle par défaut.

Le diagramme suivant illustre cette configuration:

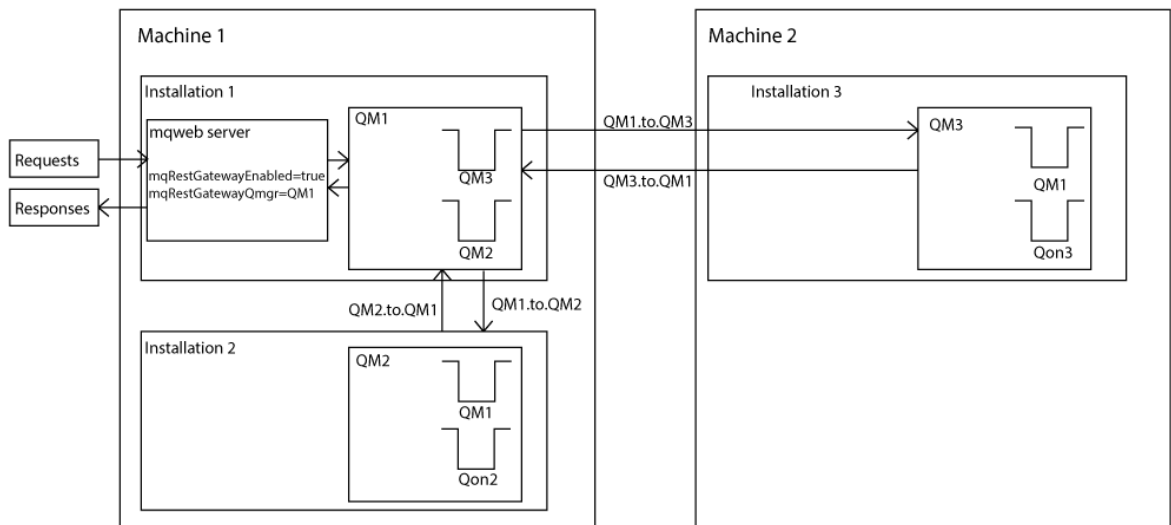


Figure 14. Diagramme d'un exemple de configuration pour l'administration à distance à l'aide de REST API.

La demande REST suivante est envoyée au serveur mqweb:

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

La réponse suivante est reçue:

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

La demande REST suivante est envoyée au serveur mqweb:

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

La réponse suivante est reçue:

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

```
    },
    "name" : "Qon3",
    "type" : "local"
  }
}
```

## REST API Horodatages

Lorsque les informations de date et d'heure sont renvoyées par le administrative REST API, elles sont renvoyées en temps universel coordonné (UTC) et dans un format défini.

La date et l'heure sont renvoyées dans le format d'horodatage suivant:

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

Par exemple, 2012-04-23T18:25:43.000Z, où Z indique que le fuseau horaire est le temps universel coordonné (UTC).

La précision de cet horodatage n'est pas garantie. Par exemple, si le serveur mqweb n'est pas démarré dans le même fuseau horaire que le gestionnaire de files d'attente spécifié dans l'URL de la ressource, l'horodatage peut ne pas être exact. De plus, si des ajustements de l'heure d'été sont nécessaires, l'horodatage peut ne pas être exact.

## REST API traitement des erreurs

Le REST API signale les erreurs en renvoyant un code de réponse HTTP approprié, par exemple 404 (Introuvable) et une réponse JSON. Tout code de réponse HTTP qui n'est pas compris entre 200 et 299 est considéré comme une erreur.

### Format de la réponse d'erreur

La réponse est au format JSON avec le codage UTF-8 . Il contient des objets JSON imbriqués:

- Objet JSON externe qui contient un tableau JSON unique appelé `error`.
- Chaque élément du tableau est un objet JSON qui représente des informations sur une erreur. Chaque objet JSON contient les propriétés suivantes:

#### **type**

Chaîne.

Type d'erreur.

#### **messageId**

Chaîne.

Identificateur unique du message de la forme MQWBnnnnX. Cet identificateur comporte les éléments suivants:

#### **MQWB**

Préfixe indiquant que le message provient de l'API REST IBM MQ .

#### **nnnn**

Numéro unique qui identifie le message.

#### **X**

Lettre unique indiquant la gravité du message:

- I si un message est purement informatif.
- W si un message signale un problème.
- E si un message indique qu'une erreur s'est produite.
- S si un message indique qu'une erreur grave s'est produite.

#### **Message**

Chaîne.

Description de l'erreur.

### explication

Chaîne.


Explication de l'erreur.

### action

Chaîne.

Description des étapes pouvant être effectuées pour résoudre l'erreur.

### qmgrName

 Cette zone est disponible uniquement pour z/OS, où le gestionnaire de files d'attente est membre du groupe de partage de files d'attente. Vous devez avoir spécifié le paramètre de requête facultatif **commandScope** ou l'attribut **queueSharingGroupDisposition**.

Chaîne.

Nom du gestionnaire de files d'attente qui a rencontré l'erreur.

Cette zone n'est pas applicable à messaging REST API.

### completionCode

Cette zone n'est disponible que si **type** est `pcf`, `java` ou `rest`.

Nombre.

Code achèvement MQ associé à l'échec.

### reasonCode

Cette zone n'est disponible que si **type** est `pcf`, `java` ou `rest`.

Nombre.

Code anomalie MQ associé à l'échec.

### exceptions

Cette zone n'est disponible que si **type** est `java`.

Tableau.

Un tableau de chaînes d'exceptions Java ou JMS. Chaque élément du tableau des exceptions contient un tableau de chaînes **stackTrace**.

Le tableau de chaînes **stackTrace** contient les détails de chaque exception divisée en lignes.

## Erreurs liées aux groupes de partage de files d'attente



Dans un groupe de partage de files d'attente, il est possible de spécifier un paramètre de requête facultatif **commandScope** pour certaines commandes. Ce paramètre permet de propager la commande à d'autres gestionnaires de files d'attente du groupe de partage de files d'attente. L'une de ces commandes peut échouer indépendamment, ce qui entraîne la réussite de certaines commandes et l'échec de certaines commandes pour le groupe de partage de files d'attente.

Dans les cas où une commande échoue partiellement, le code d'erreur HTTP 500 est renvoyé. Pour chaque gestionnaire de files d'attente qui a généré un échec, les informations sur cet échec sont renvoyées en tant qu'élément dans le tableau JSON `error`. Pour chaque gestionnaire de files d'attente qui a correctement exécuté la commande, le nom du gestionnaire de files d'attente est renvoyé en tant qu'élément dans un tableau JSON `success`.

## Exemples

- L'exemple suivant illustre la réponse d'erreur à une tentative d'obtention d'informations sur un gestionnaire de files d'attente qui n'existe pas:

```
"error": [  
  {  
    "type": "rest",  
    "messageId": "MQWB0009E",  
    "message": "MQWB0009E: Could not query the queue manager 'QM1'",
```



```

    "explanation": "The MQ REST API was invoked specifying a queue manager name which
cannot be located.",
    "action": "Resubmit the request with a valid queue manager name or no queue manager
name, to retrieve a list of queue managers. "
  }
]

```

- z/OS
 L'exemple suivant illustre la réponse d'erreur à une tentative de suppression d'une file d'attente dans un groupe de partage de files d'attente qui n'existe pas pour certains gestionnaires de files d'attente:

```

"error" : [
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason
code is 3312 : 'MQRCCF_UNKNOwn_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be
located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name
to retrieve a list of queues.",
    "qmgrName": "QM1"
  },
  {
    "type": "rest",
    "messageId": "MQWB0037E",
    "message": "MQWB0037E: Could not find the queue 'missingQueue' - the queue manager reason
code is 3312 : 'MQRCCF_UNKNOwn_OBJECT_NAME'",
    "explanation": "The MQ REST API was invoked specifying a queue name which cannot be
located.",
    "action": "Resubmit the request with the name of an existing queue, or with no queue name
to retrieve a list of queues.",
    "qmgrName": "QM2"
  }
],
"success" : [{ "qmgrName": "QM3" }, { "qmgrName": "QM4" }]

```

## Erreurs liées aux demandes MFT

Si les services MFT REST API ne sont pas activés et que vous appelez le MFT REST API, vous recevez l'exception suivante:

```

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

```

Si les services MFT REST API sont activés et que le gestionnaire de files d'attente de coordination n'est pas défini dans le fichier mqwebuser.xml, vous recevez l'exception suivante:

```

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

```

## REST API découverte

La documentation de REST API est disponible dans le IBM Documentation et au format Swagger. Swagger est une approche couramment utilisée pour documenter les API REST. La documentation Swagger de REST API peut être affichée en activant la fonction de reconnaissance d'API (apiDiscovery) sur le serveur mqweb.

### Avant de commencer

Stabilized

**Important :** La fonction apiDiscovery a été stabilisée. Vous pouvez toujours utiliser cette fonction. Actuellement, IBM MQ ne prend pas en charge l'utilisation de la fonction mpOpenAPI .

Vous devez activer la sécurité du serveur mqweb pour afficher la documentation Swagger à l'aide d'API Discovery. Pour plus d'informations sur les étapes requises pour activer la sécurité, voir [Sécurité IBM MQ Console et REST API](#).

### Procédure

1. Localisez le fichier mqwebuser.xml dans l'un des répertoires suivants:

- ALW `MQ_DATA_PATH/web/installations/installationName/servers/mqweb`
- z/OS `WLP_user_directory/servers/mqweb`

Où `WLP_user_directory` est le répertoire qui a été spécifié lors de l'exécution du script `crtmqweb` pour créer la définition de serveur mqweb.

2. Ajoutez le code XML approprié au fichier mqwebuser.xml :

- Si les balises `<featureManager>` existent dans votre fichier mqwebuser.xml , ajoutez le code XML suivant dans les balises `<featureManager>` :
- Si les balises `<featureManager>` n'existent pas dans votre fichier mqwebuser.xml , ajoutez le code XML suivant dans les balises `<server>` :

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

3. Affichez la documentation Swagger à l'aide de l'une des méthodes suivantes:

- Affichez une page Web que vous pouvez parcourir et essayer REST API en entrant l'URL suivante dans un navigateur:

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

Outre l'authentification de chaque demande, vous devez inclure un en-tête `ibm-mq-rest-csrf-token` pour chaque demande POST, PATCH ou DELETE. Le contenu de cet en-tête peut être n'importe quelle chaîne, y compris vide.

Cet en-tête de demande est utilisé pour confirmer que les données d'identification utilisées pour authentifier la demande sont utilisées par le propriétaire des données d'identification. C'est-à-dire que le jeton est utilisé pour empêcher les attaques de falsification de requêtes entre sites.

- Extrayez un document Swagger 2 unique qui décrit l'ensemble de REST API en émettant une requête HTTP GET à l'URL suivante:

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

Ce document peut être utilisé pour les applications dans lesquelles vous souhaitez naviguer à l'aide d'un programme dans les API disponibles.

**host**

Indique le nom d'hôte ou l'adresse IP sur lequel REST API est disponible.

La valeur par défaut est localhost.

**port**

Indique le numéro de port HTTPS utilisé par administrative REST API .

La valeur par défaut est 9443.

Si le nom d'hôte ou le numéro de port par défaut est modifié, vous pouvez déterminer les valeurs correctes à partir de l'URL REST API . Utilisez la commande **dspmweb status** pour afficher l'URL.

**Information associée**

[dspmweb status \(affichage de l'état du serveur mqweb\)](#)

## REST APISupport de langue nationale (NLS)

REST API prend en charge, avec certaines qualifications, la possibilité de spécifier des langues nationales dans le cadre d'une demande HTTP.

### Arrière-plan

Les [en-têtes HTTP](#) permettent de spécifier un comportement particulier sur les demandes et de fournir des informations supplémentaires dans les réponses.

Les en-têtes HTTP incluent la possibilité de demander que les informations soient renvoyées dans une langue nationale. Le REST API respecte cet en-tête dans la mesure du possible.

### Spécification d'une langue nationale

Dans l'en-tête HTTP ACCEPT-LANGUAGE, une ou plusieurs balises de langue peuvent être fournies. Vous pouvez éventuellement associer un rang aux balises, ce qui permet de spécifier une liste triée par préférence. [Cette page](#) contient une discussion utile sur le principe.

REST API respecte cet en-tête, en sélectionnant une langue dans l'en-tête ACCEPT-LANGUAGE et en renvoyant des messages dans cette langue. Lorsque l'en-tête ACCEPT-LANGUAGE ne contient aucune langue que le REST API peut prendre en charge, les messages sont renvoyés dans une langue par défaut. Cette langue par défaut correspond à l'environnement local par défaut du serveur Web REST API .

La section [«Quelles données sont traduites?»](#), à la page 91 explique les données qui sont traduites.

### Indication de la langue applicable dans les réponses

L'en-tête HTTP CONTENT-LANGUAGE sur les réponses du REST API indique la langue dans laquelle les messages sont renvoyés.

### Quelles données sont traduites?

Les messages d'erreur et d'information sont traduits ; les autres textes ne sont pas traduits.

- Les données renvoyées par un gestionnaire de files d'attente ne sont pas traduites. Par exemple, dans le cas de l'exécution d'une commande MQSC via REST API, les réponses du gestionnaire de files d'attente sont dans l'environnement local du gestionnaire de files d'attente.
- La documentation générée (Swagger) pour le REST API, telle qu'exposée via la fonction `apiDiscovery`, est en anglais.

### Quelles sont les langues prises en charge?

Outre l'anglais, les messages d'erreur et d'information REST API sont traduits dans les langues suivantes.

#### Chinois (simplifié)

Dénoté par la balise de langue zh\_CN

**Chinois (traditionnel)**

Dénoté par la balise de langue zh\_TW

**Tchèque**

Dénoté par la balise de langue cs

**Français**

Dénoté par la balise de langue fr

**Hongrois**

Dénoté par la balise de langue hu

**Italien**

Dénoté par la balise de langue it

**Japonais**

Dénoté par la balise de langue ja

**Coréen**

Dénoté par la balise de langue ko

**Polonais**

Dénoté par la balise de langue pl

**(Brésil) Portugais**

Dénoté par la balise de langue pt\_BR

**Russe**

Dénoté par la balise de langue ru

**Espagnol**

Dénoté par la balise de langue es

**Exemples**

Dans les exemples, le serveur Web possède un environnement local par défaut en anglais.

**Spécification d'une langue prise en charge unique**

Dans les en-têtes de demande, ACCEPT-LANGUAGE est défini sur fr. Ce paramètre indique que le français est la langue préférée pour le texte traduisible.

Dans les en-têtes de réponse, CONTENT-LANGUAGE est défini sur fr. Ce paramètre indique que les messages d'erreur et d'information de la réponse sont en français.

**Spécification d'une liste de langues**

Dans les en-têtes de demande, ACCEPT-LANGUAGE est défini sur am, fr. Ce paramètre spécifie que l'amharique et le français sont des langues acceptables pour le texte traduisible et que l'amharique est la langue préférée pour le texte traduisible.

Dans les en-têtes de réponse, CONTENT-LANGUAGE est défini sur fr. Ce paramètre indique que les messages d'erreur et d'information de la réponse sont en français, car REST API ne prend pas en charge l'amharique.

**Spécification d'une langue unique non prise en charge**

Dans les en-têtes de demande, ACCEPT-LANGUAGE est défini sur am. Ce paramètre indique que l'amharique est la langue préférée pour le texte traduisible.

Dans les en-têtes de réponse, CONTENT-LANGUAGE est défini sur en. Ce paramètre indique que les messages d'erreur et d'information de la réponse sont en anglais, car REST API ne prend pas en charge l'amharique.

**REST API Versions**

Le numéro de version REST API fait partie de l'URL de base pour les demandes REST. Par exemple, <https://localhost:9443/ibmmq/rest/v2/admin/installation>. Le numéro de version est utilisé pour isoler les clients des modifications apportées au REST API qui pourraient être introduites dans les éditions futures.

IBM MQ 9.2.0 introduit la version 2 de REST API. Cette augmentation de version s'applique à administrative REST API, messaging REST API, et MFT REST API. Cette augmentation de version change l'URL de ressource qui est utilisée pour l'REST API. Le préfixe d'URL pour les URL de ressource de la version 2 est l'URL suivante :

```
https://host:port/ibmmq/rest/v2/
```

**Stabilized** Certaines modifications apportées à la REST API peuvent modifier la fonction REST API existante de sorte que les clients qui utilisent le REST API peuvent avoir besoin d'être mis à jour. Pour éviter que de telles modifications ne forcent les clients à être mis à jour, le numéro de version REST API est augmenté et la fonction existante est stabilisée au numéro précédent. La nouvelle fonction qui peut modifier la fonction existante est ajoutée à REST API au niveau du nouveau numéro de version. Par conséquent, les clients peuvent continuer à utiliser REST API à la version précédente sans être mis à jour.

Les modifications REST API qui peuvent nécessiter une mise à jour du client incluent les modifications suivantes:

- Suppression de la prise en charge d'un attribut existant dans le JSON qui est envoyé à REST API ou renvoyé par ce dernier.
- Suppression d'une URL, d'un verbe HTTP ou d'un en-tête. Par exemple, si une URL ou un en-tête est renommé, ou si une instruction différente est utilisée.
- Ajout d'un nouvel attribut JSON obligatoire aux données envoyées à une URL existante.
- Ajout d'un nouvel en-tête HTTP obligatoire aux données envoyées à une URL existante.
- Ajout d'un nouveau paramètre de requête obligatoire à une URL existante.

Lorsque ce type de modification est introduit dans la fonction REST API qui existait dans une édition Long Term Support (LTS), le numéro de version de REST API est augmenté pour la première de ces modifications. Toutes les modifications ultérieures apportées dans une édition Continuous Delivery (CD) qui peuvent nécessiter des modifications des clients qui utilisent le REST API utilisent le nouveau numéro de version.

Ce numéro de version reste le même dans les éditions suivantes de CD jusqu'à la prochaine édition de LTS . Par conséquent, le numéro de version augmente au maximum une fois entre les éditions LTS .

**Stabilized** Lorsque le numéro de version est augmenté, la fonction REST API existante est stabilisée à l'ancien numéro de version. C'est-à-dire que la fonction REST API existante qui était disponible dans l'édition LTS reste disponible avec l'ancien numéro de version, mais aucune autre modification n'est apportée à cette version. Toute nouvelle fonction ajoutée à REST API est ajoutée à la nouvelle version de REST API . Toutefois, il n'est pas garanti que les ajouts apportés à REST API dans les éditions CD antérieures à l'augmentation de la version soient inclus dans l'ancienne version de REST API.

**Deprecated** Les clients existants peuvent continuer à utiliser le REST API à l'ancien numéro de version sans qu'aucune modification ne soit nécessaire. Les versions plus anciennes de REST API peuvent être obsolètes et éventuellement supprimées.

Certaines modifications ne nécessitent pas de modifications apportées aux clients qui utilisent le REST API. Ces modifications n'entraînent pas d'augmentation du numéro de version. Par conséquent, assurez-vous que tout client qui utilise le REST API n'a pas besoin d'être mis à jour lorsque ces types de modifications sont introduits. Ces modifications apportées à REST API peuvent inclure les modifications suivantes:

- Ajout d'un nouvel attribut JSON aux données existantes renvoyées par REST API.
- Ajout d'une nouvelle URL.
- Ajout d'une nouvelle instruction HTTP à une URL existante.
- Ajout d'un nouveau code de statut à une URL existante.
- Ajout de nouveaux attributs JSON facultatifs aux données envoyées à une URL existante.
- Ajout de nouveaux paramètres de requête sur une URL existante.

- Ajout de nouveaux en-têtes aux données envoyées à une URL existante.
- Renvoi de nouveaux en-têtes à partir de REST API.

## Modifications apportées à la nouvelle fonction d'API REST Continuous Delivery

Pour la nouvelle fonction REST API ajoutée dans une édition CD , les modifications apportées à cette nouvelle fonction qui peuvent nécessiter des modifications des clients REST API n'augmentent pas le numéro de version. Autrement dit, la nouvelle fonction peut être modifiée avant la prochaine édition de LTS sans augmenter le numéro de version. Lorsque la fonction est incluse dans une édition LTS , toutes les modifications ultérieures pouvant nécessiter des modifications des clients REST API augmentent le numéro de version.

### Exemple

1. Dans LTS édition X, REST API est à la version 1.
2. Dans CD version X.0.1, la prise en charge d'une nouvelle URL est ajoutée. Cette modification ne nécessite pas de modifications apportées aux clients qui utilisent le REST API. Par conséquent, REST API reste à la version 1.
3. Dans CD X.0.2, la prise en charge d'une nouvelle URL est ajoutée. Cette modification ne nécessite pas de modifications apportées aux clients qui utilisent l'API REST. Par conséquent, REST API reste à la version 1.
4. A LTS édition Y, REST API est à la version 1.
5. Dans CD édition Y.0.1, une URL existante est renommée. Cette modification peut nécessiter des modifications apportées aux clients qui utilisent le REST API. Par conséquent, une nouvelle version de REST API est créée en tant que version 2. L'URL renommée est incluse dans la version 2 de REST API, avec toutes les fonctions existantes. Toute nouvelle fonction ajoutée à REST API est ajoutée à la version 2. La version 1 reste stabilisée au niveau de l'édition Y de LTS .
6. Dans CD édition Y.0.2, une autre URL existante est renommée. Comme la version est déjà augmentée dans l'édition Y de CD , REST API reste à la version 2. La version 1 reste stabilisée au niveau de l'édition Y de LTS .
7. A l'édition Z de LTS , REST API reste à la version 2. La version 1 reste stabilisée au niveau de l'édition Y de LTS .

## Administration à l'aide de IBM MQ Console

---

Vous pouvez effectuer des tâches d'administration de base à l'aide de l' IBM MQ Console.

**Remarque :** Ne désactivez pas le serveur de commandes sur vos gestionnaires de files d'attente lorsque vous utilisez IBM MQ Console. Si le serveur de commandes est désactivé pour un gestionnaire de files d'attente:

- Le IBM MQ Console ne répond plus, avec de longs retards dans le traitement des commandes
- Le délai d'attente des commandes émises au gestionnaire de files d'attente est écoulé.

### Tâches associées

[Traçage du IBM MQ Console](#)

## Guide d'initiation à IBM MQ Console

Configurez le serveur mqweb ; déterminez l'URI du IBM MQ Console; connectez-vous à la console ; connectez-vous à la console.

### Avant de commencer

Pour effectuer cette tâche, vous devez être un utilisateur doté de certains privilèges vous permettant d'employer la commande **[dspmqweb](#)** :

- ▶ **z/OS** Sur z/OS, vous devez disposer des droits permettant d'exécuter la commande **dspmweb** et de l'accès en écriture sur le fichier `mqwebuser.xml`.
  - ▶ **Multi** Sur tous les autres systèmes d'exploitation, vous devez être un [utilisateur privilégié](#).
- IBM i** Sous IBM i, les commandes doivent être exécutées dans QSHELL.

## Pourquoi et quand exécuter cette tâche

Tenez compte des restrictions suivantes :

- ▶ **z/OS**
  - Les gestionnaires de files d'attente sur z/OS ne peuvent pas être créés, supprimés, démarrés ou arrêtés.
  - Les initiateurs de canal sous z/OS ne peuvent pas être démarrés ou arrêtés et le statut de l'initiateur de canal n'est pas affiché.
  - Les programmes d'écoute ne peuvent pas être affichés ou administrés.
  - Les commandes de canal de démarrage, de commande ping, de résolution et de réinitialisation ne peuvent être émises qu'avec CHLDISP (DEFAULT).
  - Les objets définis avec QSGDISP (GROUP) ne peuvent pas être affichés ou gérés.
  - La sécurité du gestionnaire de files d'attente ne peut pas être gérée.
  - L'utilisation des ressources système ne peut pas être surveillée.
- ▶ **Multi**
  - Vous ne pouvez pas utiliser IBM MQ Console pour utiliser des canaux AMQP.
  - Vous ne pouvez pas utiliser IBM MQ Console pour utiliser les canaux MQTT.

## Procédure

1. Si le serveur mqweb n'est pas déjà configuré pour être utilisé par IBM MQ Console, configurez le serveur mqweb.

Pour plus d'informations sur la configuration du serveur mqweb avec un registre de base, voir [Configuration de base pour le serveur mqweb](#).

2. ▶ **z/OS**

Sur z/OS, définissez la variable d'environnement `WLP_USER_DIR` afin de pouvoir utiliser la commande **dspmweb**. Définissez la variable de telle sorte qu'elle pointe vers votre configuration de serveur mqweb à l'aide de la commande suivante :

```
export WLP_USER_DIR=WLP_user_directory
```

où `WLP_user_directory` est le nom du répertoire transmis à `crtmqweb`. Par exemple :

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

Pour plus d'informations, voir [Creating the mqweb server](#).

3. Déterminez l'URI de IBM MQ Console en entrant la commande suivante:

```
dspmweb status
```

La commande génère une sortie similaire à la suivante:

```
MQWB1124I: Server 'mqweb' is running.
URLS:
https://localhost:9443/ibmmq/rest/v1/
https://localhost:9443/ibmmq/console/
```

L'URI de IBM MQ Console se termine par le suffixe console/.

4. Connectez-vous à IBM MQ Console en entrant l'URL de l'étape précédente dans un navigateur.

Une exception de sécurité peut être générée par le navigateur car le certificat par défaut fourni avec le serveur mqweb n'est pas un certificat digne de confiance. Choisissez de passer à l' IBM MQ Console.

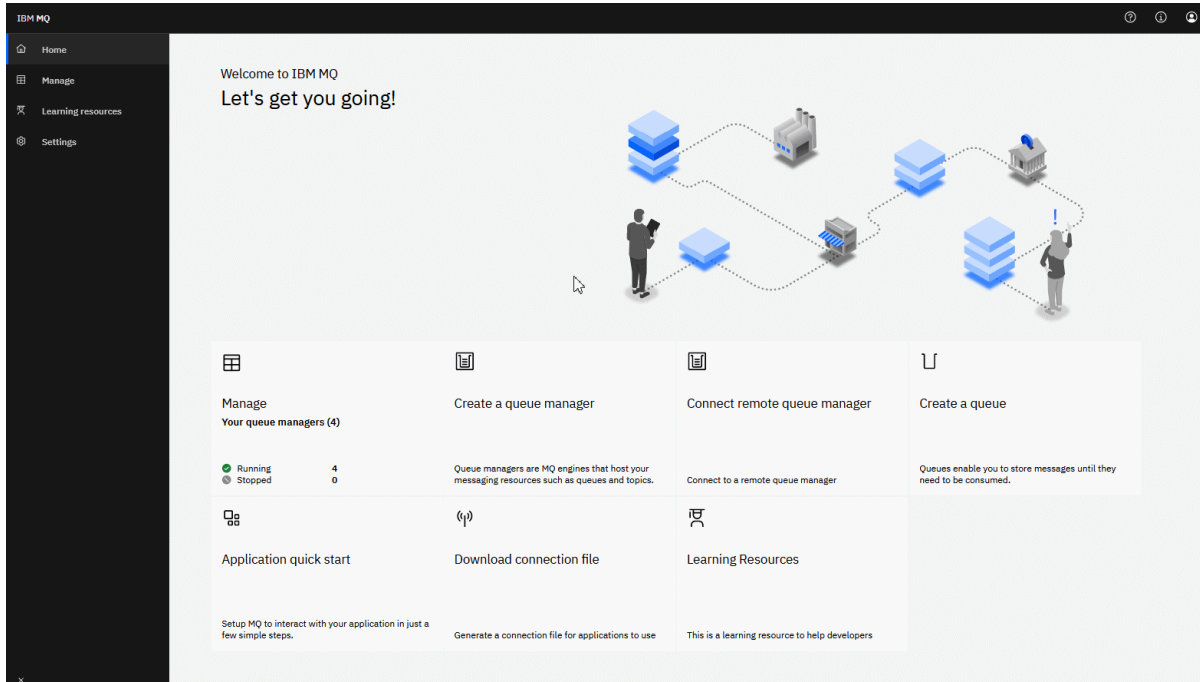
5. Connectez-vous à IBM MQ Console. Utilisez le nom d'utilisateur mqadmin et le mot de passe mqadmin.

## Que faire ensuite

Par défaut, IBM MQ Console utilise l'authentification basée sur un jeton pour authentifier les utilisateurs. Vous pouvez également utiliser l'authentification par certificat client. Pour plus d'informations, voir [Utilisation de l'authentification par certificat client avec REST API et IBM MQ Console](#).

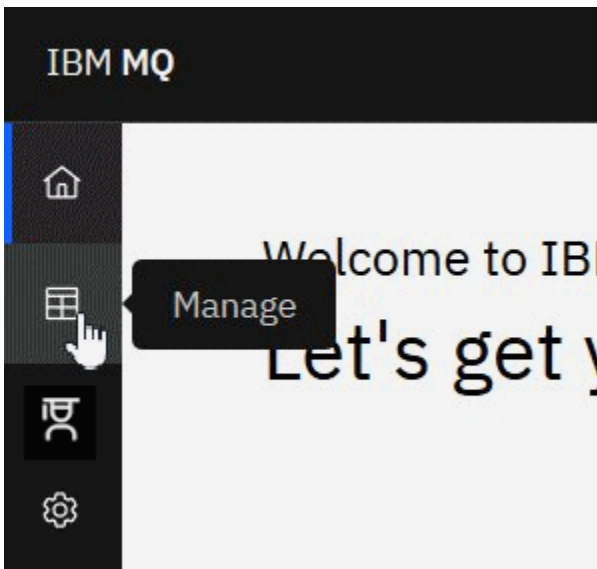
## V 9.4.0 Présentation rapide du IBM MQ Console

Lorsque vous vous connectez pour la première fois à IBM MQ Console, vous accédez à la page d'arrivée. A partir de là, vous pouvez choisir de gérer des gestionnaires de files d'attente existants, de créer un gestionnaire de files d'attente ou une file d'attente, d'accéder à certaines rubriques de formation ou d'ouvrir les informations produit IBM MQ dans IBM Documentation. Vous pouvez également lancer le démarrage rapide de l'application, qui vous guide tout au long du processus de configuration rapide et facile de la messagerie entre les applications et les gestionnaires de files d'attente nouveaux ou existants.

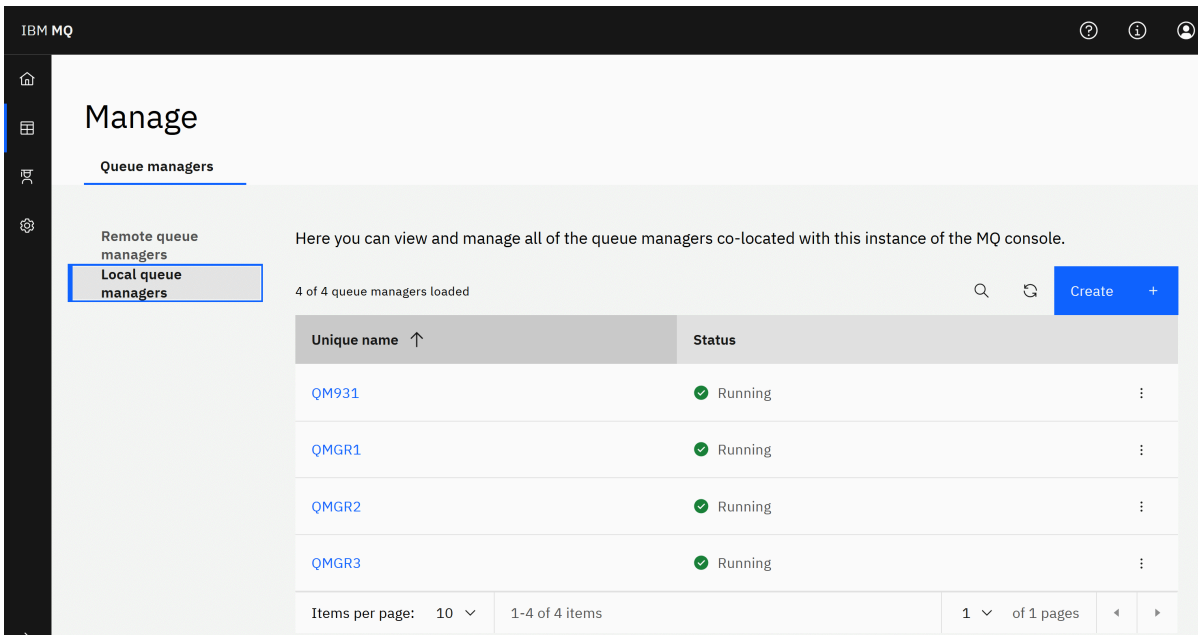


Vous pouvez également cliquer sur l'icône de gestion pour commencer à gérer les objets IBM MQ immédiatement.

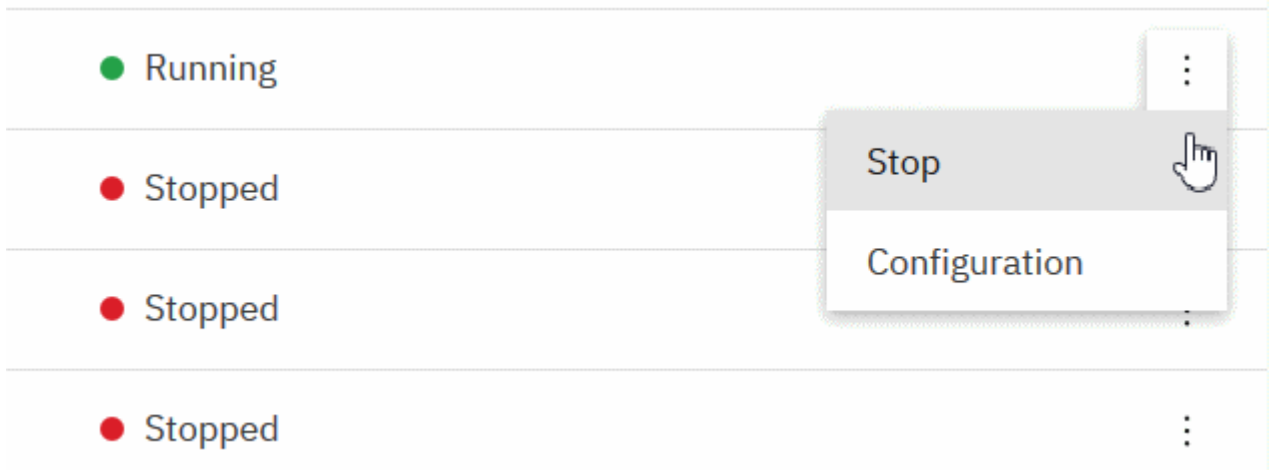




La vue de gestion affiche initialement les gestionnaires de files d'attente et leur état en cours. Vous pouvez également créer de nouveaux gestionnaires de files d'attente et connecter des gestionnaires de files d'attente éloignées.

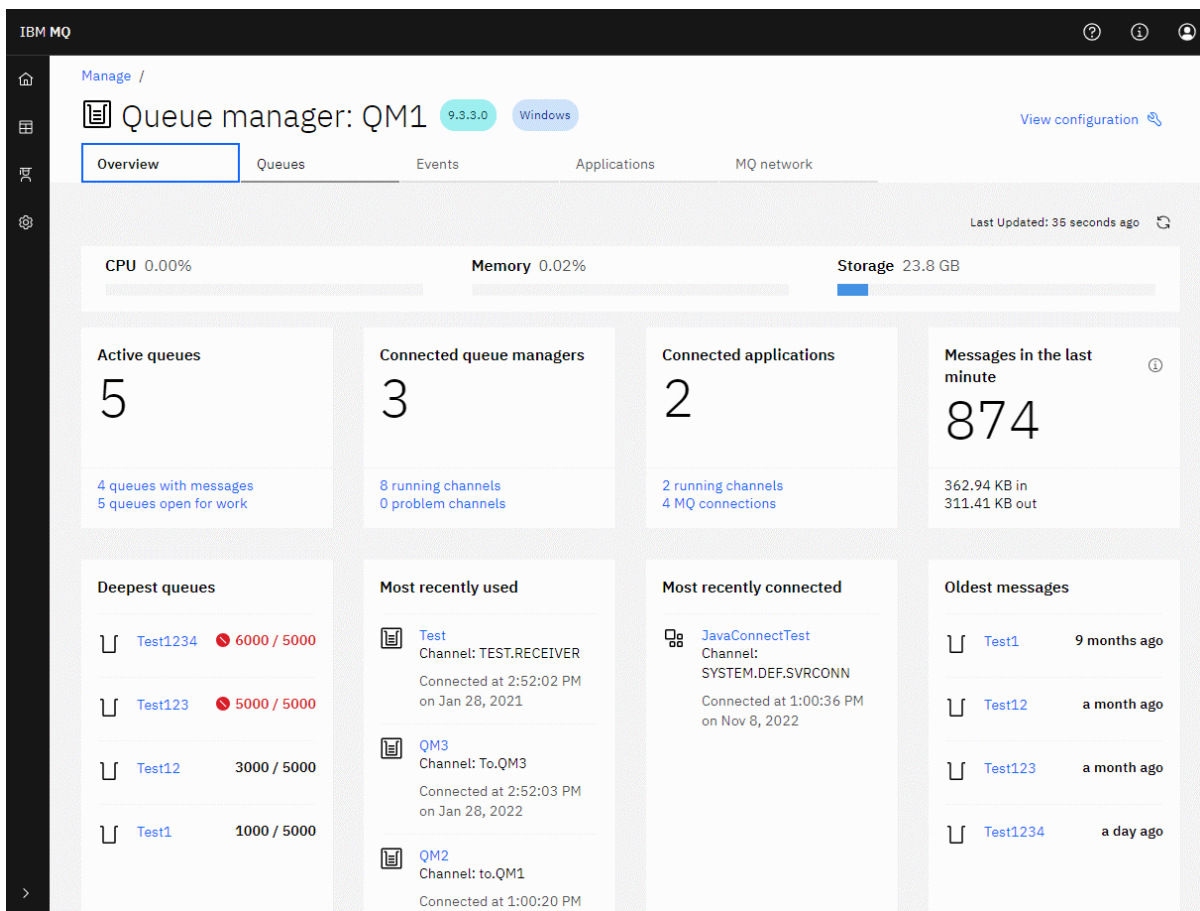


Chaque gestionnaire de files d'attente dispose d'un menu qui vous permet d'arrêter ou de configurer un gestionnaire de files d'attente en cours d'exécution, ou de démarrer ou de supprimer un gestionnaire de files d'attente arrêté.



Les enregistrements de droits d'accès, les objets Informations d'authentification et les enregistrements d'authentification de canal pour le gestionnaire de files d'attente sont disponibles dans l'onglet **Sécurité** de la page **Configuration** du gestionnaire de files d'attente, où vous pouvez en créer et en ajouter de nouveaux.

Cliquez sur le nom d'un gestionnaire de files d'attente en cours d'exécution pour ouvrir son tableau de bord.



A partir du tableau de bord du gestionnaire de files d'attente, vous pouvez effectuer les actions suivantes:

**V 9.4.0** Dans l'onglet **Présentation**, affichez les informations suivantes:

## **UC**

Estimation en pourcentage de l'utilisation de l'unité centrale par le gestionnaire de files d'attente. (Non applicable sous z/OS.)

## **Mémoire**

Estimation en pourcentage de l'utilisation de la mémoire par le gestionnaire de files d'attente. (Non applicable sous z/OS ou Windows.)

## **Mémoire**

Estimation en pourcentage de l'espace disponible du disque sur lequel réside le gestionnaire de files d'attente. (Non applicable sous z/OS.)

## **Files d'attente actives**

Nombre de files d'attente contenant des messages ou ouvertes en entrée ou en sortie.

## **Gestionnaires de files d'attente connectés**

Nombre de gestionnaires de files d'attente actuellement connectés, tel que dérivé des canaux actifs.

## **Applications connectées**

Nombre d'applications actuellement connectées.

## **Messages de la dernière minute**

Affiche un récapitulatif des rubriques système PUT/GET qui affichent le débit des messages toutes les 10 secondes. (Non applicable sous z/OS.)

## **Abonnements**

Affiche le nombre d'abonnements. Visible uniquement sur z/OS et sur les autres plateformes sur lesquelles la surveillance des rubriques système est désactivée (voir [setmqweb properties](#)).

## **Plus grandes files d'attente**

Répertorie les files d'attente par ordre de longueur. Affiche la longueur en cours et la longueur maximale de la file d'attente.

## **Utilisé le plus récemment**

Répertorie les gestionnaires de files d'attente actuellement connectés, classés par date de dernier message.

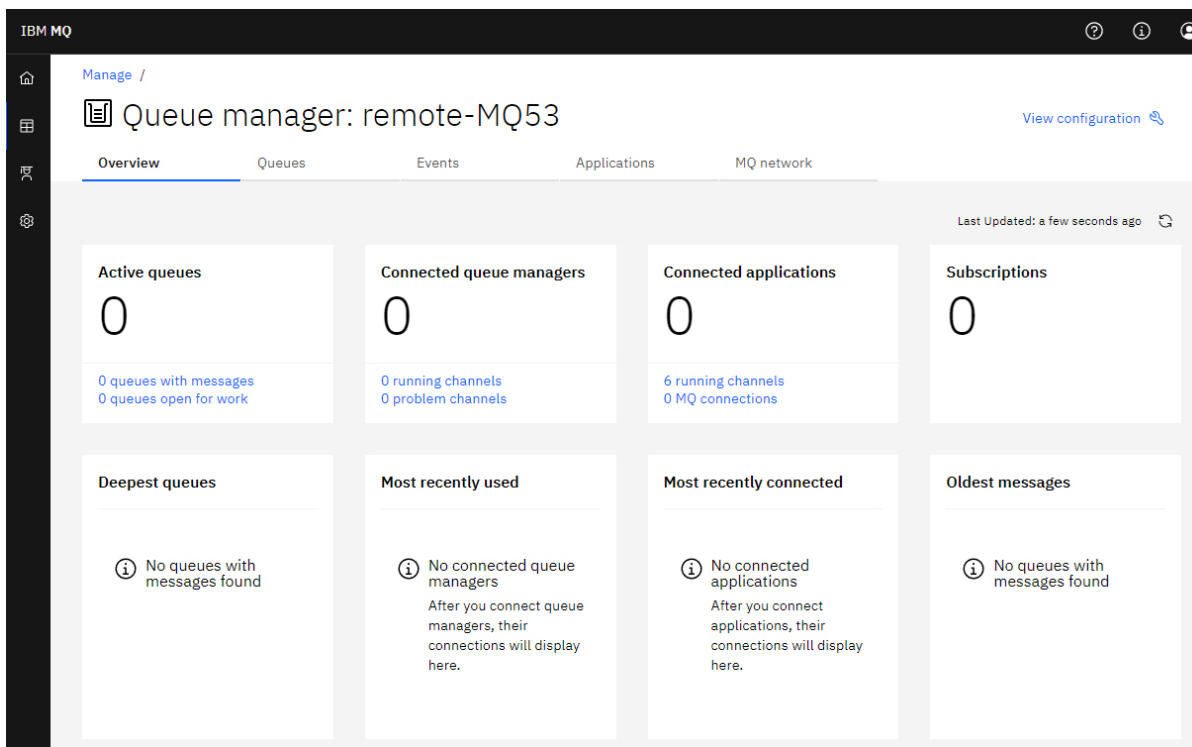
## **Le plus récemment connecté**

Répertorie les applications actuellement connectées en tant que dérivées des canaux de connexion serveur actifs, classées par date et heure de début de canal.

## **Messages les plus anciens**

Répertorie les files d'attente classées en fonction de la date et de l'heure du message le plus ancien.

Les informations affichées dans l'onglet **Présentation** sont dérivées de la surveillance des rubriques système (voir [Métriques publiées sur les rubriques système](#)). z/OS ne prend pas en charge la surveillance des rubriques système et la surveillance à des fins d'affichage de la console peut être désactivée sur d'autres plateformes (voir [setmqweb properties](#)). Dans ces cas, l'onglet **Présentation** affiche des informations plus limitées et son apparence est similaire à l'exemple suivant:




Dans l'onglet **Files d'attente** :


- Créer des files d'attente
- Cliquez sur un nom de file d'attente pour afficher les messages existants et en créer de nouveaux, et pour configurer la file d'attente.

Dans l'onglet **Événements** :

### Rubriques

- Créer de nouvelles rubriques
- Configurer les rubriques existantes 
- Cliquez sur un nom de rubrique pour afficher les abonnements correspondants

### Abonnements

- Créer de nouveaux abonnements gérés ou non gérés
- Configurer les abonnements existants 

**V 9.4.0** Dans l'onglet **Applications** :

### Présentation

Contient des vignettes qui fournissent des vues d'ensemble des statistiques suivantes:

#### Applications connectées

Affiche le nombre d'applications connectées. Fournit des liens vers les onglets suivants:

- **Instances d'application**
- **Connexions**

#### Exécution d'instances de canal

Affiche un comptage du nombre d'instances de canal SVRCONN et de ces liens vers celles qui sont définies ou arrêtées dans l'onglet **Canal d'application** .

### **Connexions**

Affiche le nombre de connexions. Fournit des liens vers les informations suivantes dans l'onglet **Connexions** :

- Connexions locales (sans nom de canal)
- Connexions distantes (celles avec un nom de canal)

### **Applications les plus courantes**

Affiche une liste des applications fréquentes, classées en fonction du nombre de connexions qu'elles utilisent.

### **Canaux les plus courants**

Affiche la liste des canaux fréquents, classés par nombre d'instances actives.

### **Transactions les plus anciennes**

Affiche la liste des transactions les plus anciennes par nom d'application. Ces transactions ont des connexions avec des unités d'oeuvre ouvertes et sont classées en fonction de la date et de l'heure de début de l'unité de travail.

### **Versions connectées à distance**

Affiche la liste des versions IBM MQ connectées communes, c'est-à-dire les instances de canal ayant une REMOTE\_VERSION spécifiée.

### **Sécurité du canal d'application**

Affiche la liste des protocoles de sécurité de canal connectés communs, c'est-à-dire les instances de canal ayant un protocole SECURITY\_PROTOCOL spécifié.

### **Débits de transfert de canal**

Affiche la liste des canaux communs classés en fonction des débits de transfert des messages et des octets. Utilise la date et l'heure de début du canal pour calculer la durée et utilise MSGS et MQIACH\_BYTES\_SENT/ MQIACH\_BYTES\_RCVD pour calculer le taux.

## **Applications**

Affichez des informations sur les applications connectées au gestionnaire de files d'attente.

### **Canaux**

Afficher l'activité sur les canaux connectés aux applications.

### **Canaux d'application**

- Démarrage, arrêt, commande ping et configuration des canaux
- Créer de nouveaux canaux
- Réinitialiser les canaux

### **Instances de canal d'application**

- Afficher le statut des instances de canal d'application
- Résoudre les messages en attente de validation sur les canaux

**V 9.4.0**

Dans l'onglet **Réseau MQ** :

### **Présentation**

Contient des vignettes qui fournissent des vues d'ensemble des statistiques suivantes:

#### **Instances en cours d'exécution de canal de gestionnaire de files d'attente**

Affiche le nombre d'instances de canal non SVRCONN. Affiche des liens vers les types d'instance de canal suivants dans l'onglet **Gestionnaires de files d'attente connectés** :

- Canaux définis
- Canaux arrêtés

### **Gestionnaires de files d'attente connectés**

Affiche le nombre de gestionnaires de files d'attente connectés par MQCA\_REMOTE\_Q\_MGR\_NAME. Fournit également le nombre de gestionnaires de files d'attente renvoyés par MQCMD\_INQUIRE\_CLUSTER\_Q\_MGR.

### **Appartenance au(x) cluster(s)**

S'il n'existe qu'un seul cluster de gestionnaires de files d'attente, affiche le nom du cluster et indique si le gestionnaire de files d'attente est un référentiel complet ou partiel. Affiche le nombre de gestionnaires de files d'attente visibles dans le cluster. S'il existe plusieurs clusters, affiche le nombre de clusters plus un nombre de gestionnaires de files d'attente de référentiel complet et partiel dans chaque cluster.

### **Canaux e gestionnaire de files d'attente défailants**

Affiche la liste des canaux qui sont à l'état de relance (non arrêté/en cours d'exécution). Calcule le nombre de nouvelles tentatives restantes si elles sont à l'état de relance. La liste contient des canaux avec les types de statut suivants:

- MQCHS\_PAUSED
- MQCHS\_RELANCE

### **Délais de message les plus longs**

Affiche la liste des canaux ayant un indicateur de temps XMIT (longue période).

### **Files d'attente de transmission sans surveillance**

Affiche la liste des files d'attente de transmission dont la longueur de la file d'attente est différente de zéro et qui ne sont pas associées à des descripteurs.

### **Versions de connexion distante**

Affiche la liste des versions IBM MQ connectées communes, c'est-à-dire les instances de canal ayant une REMOTE\_VERSION spécifiée.

### **Sécurité du canal du gestionnaire de files d'attente**

Affiche la liste des protocoles de sécurité de canal connectés communs, c'est-à-dire les instances de canal ayant un protocole SECURITY\_PROTOCOL spécifié.

### **État de santé du cluster**

Affiche un certain nombre de statistiques indépendantes relatives à la santé du cluster. Le statut inclut:

- Nombre d'objets de cluster (files d'attente, rubriques, gestionnaires de files d'attente).
- Nombre de gestionnaires de files d'attente interrompus (MQIACF\_SUSPEND défini sur YES).
- Profondeur du système SYSTEM.CLUSTER.COMMAND.QUEUE .
- Nombre d'entrées de gestionnaire de files d'attente de cluster commençant par SYSTEM.TEMP.

Si toutes ces valeurs sont nulles, cette vignette n'est pas affichée et la vignette **Programmes d'écoute** est affichée à la place.


### **Programmes d'écoute**

Affiche la liste des programmes d'écoute et indique s'ils sont en cours d'exécution. S'affiche uniquement si la vignette **Santé du cluster** n'est pas affichée.


### **Gestionnaires de files d'attente connectés**

Affichez les détails des gestionnaires de files d'attente actuellement connectés à ce gestionnaire de files d'attente.


### **Canaux de gestionnaire de file d'attente**

- Démarrage, arrêt, commande ping et configuration des canaux 
- Créer de nouveaux canaux
- Réinitialiser les canaux


## Instances de canal de gestionnaire de files d'attente


- Afficher le statut des instances de canal de gestionnaire de files d'attente
- Résoudre les messages en attente de validation sur les canaux 

## IBM MQ Console: Utilisation des gestionnaires de files d'attente locales

Vous pouvez créer, configurer et contrôler des gestionnaires de files d'attente locales à partir du niveau supérieur de la vue Gestion. 

### Pourquoi et quand exécuter cette tâche

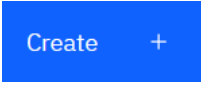


 La vue Gestion répertorie les gestionnaires de files d'attente locaux qui sont ajoutés à l'installation IBM MQ à partir de laquelle IBM MQ Console s'exécute. Les gestionnaires de files d'attente associés à différentes installations de IBM MQ sur le même système ne sont pas répertoriés.




 Sous z/OS, la vue Gestion répertorie les gestionnaires de files d'attente dont la version est identique à celle d' IBM MQ Console et qui sont définis sur le système sur lequel IBM MQ Console s'exécute. Les gestionnaires de files d'attente d'une version différente de IBM MQ Console ne sont pas répertoriés.

Vous pouvez sélectionner des gestionnaires de files d'attente individuels dans la liste.

**Remarque :** IBM MQ Console peut se connecter à un gestionnaire de files d'attente de données répliquées (RDQM) local lorsqu'il est actif (c'est-à-dire qu'il a le rôle principal), mais n'offre pas de fonctions spécifiques aux gestionnaires de files d'attente de données répliquées.

### Procédure

- Pour créer un gestionnaire de files d'attente local:
  - a) Cliquez sur le bouton Créer  dans la vue de liste des gestionnaires de files d'attente.
  - b) Entrez un nom pour le nouveau gestionnaire de files d'attente. Vous pouvez indiquer 48 caractères au maximum. Les caractères valides sont les lettres, les chiffres, ainsi que les caractères ".", "/", "\_", et "%".
  - c) Facultatif : Entrez un port TCP/IP disponible sur lequel le gestionnaire de files d'attente sera à l'écoute. Le numéro de port ne doit pas dépasser 65535.
  - d) Cliquez sur **Créer**. Le nouveau gestionnaire de files d'attente est créé et démarré.
- Pour démarrer un gestionnaire de files d'attente local:
  - a) Localisez le gestionnaire de files d'attente que vous souhaitez démarrer dans la liste.
  - b) Sélectionnez **Démarrer** dans le menu .
- Pour arrêter un gestionnaire de files d'attente local:
  - a) Sélectionnez le gestionnaire de files d'attente à arrêter dans la liste du widget de gestionnaire de files d'attente local.
  - b) Sélectionnez **Arrêter** dans le menu .
- Pour supprimer un gestionnaire de files d'attente local:

- a) Si le gestionnaire de files d'attente est en cours d'exécution, arrêtez-le.
- b) Sélectionnez **Afficher la configuration** dans le menu  et sélectionnez **Supprimer le gestionnaire de files d'attente**.
- c) Confirmez la suppression du gestionnaire de files d'attente en entrant son nom dans la fenêtre de confirmation. Le gestionnaire de files d'attente et tous les objets associés sont supprimés.
- Pour afficher et éditer les propriétés d'un gestionnaire de files d'attente local:
  - a) Vérifiez que le gestionnaire de files d'attente est en cours d'exécution et localisez-le dans la liste des gestionnaires de files d'attente.
  - b) Sélectionnez **Afficher la configuration** dans le menu  .
  - c) Vérifiez que l'onglet **Propriétés** est sélectionné. Affichez les propriétés et éditez-les si nécessaire. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande. Pour plus d'informations sur une propriété, vous pouvez l'afficher dans [Propriétés du gestionnaire de files d'attente](#).
- Pour utiliser les paramètres de sécurité du gestionnaire de files d'attente local:
  - a) Assurez-vous que le gestionnaire de files d'attente est en cours d'exécution et sélectionnez-le dans la liste des gestionnaires de files d'attente.
  - b) Sélectionnez **Afficher la configuration** dans le menu  .
  - c) Vérifiez que l'onglet **Sécurité** est sélectionné.
  - d) Vous pouvez utiliser des objets d'authentification, des enregistrements d'autorisation ou des objets d'authentification de canal. Pour plus d'informations, consultez les rubriques suivantes:
    - [«IBM MQ Console: Utilisation des objets d'informations d'authentification»](#), à la page 104
    - [«IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente»](#), à la page 106
    - [«IBM MQ Console: Utilisation des enregistrements d'authentification de canal»](#), à la page 107

#### **V 9.4.0 IBM MQ Console: Utilisation des objets d'informations d'authentification**

Vous pouvez utiliser la console pour ajouter et supprimer des objets d'informations d'authentification sur un gestionnaire de files d'attente. Vous pouvez également afficher et définir les propriétés et gérer les enregistrements de droits d'accès pour les objets.

### **Pourquoi et quand exécuter cette tâche**


La vue des informations d'authentification répertorie les informations d'authentification qui existent pour un gestionnaire de files d'attente spécifique. Vous pouvez sélectionner des informations d'authentification individuelles dans la liste à utiliser.

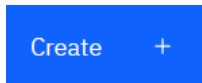
Les informations d'authentification du gestionnaire de files d'attente font partie du support IBM MQ pour le protocole TLS (Transport Layer Security). Ces objets contiennent les définitions requises pour effectuer la vérification de la révocation de certificat à l'aide d'OCSP ou de listes de révocation de certificat (CRL) sur les serveurs LDAP, ainsi que les définitions requises pour activer la vérification de l'ID utilisateur et du mot de passe.

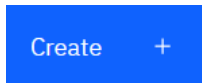
### **Procédure**

- Pour afficher les informations d'authentification d'un gestionnaire de files d'attente:
  - a) Assurez-vous que le gestionnaire de files d'attente est en cours d'exécution et sélectionnez-le dans la liste des gestionnaires de files d'attente.




- b) Sélectionnez **Afficher la configuration** dans le menu  .
  - c) Vérifiez que l'onglet **Sécurité** est sélectionné.
  - d) Sélectionnez **Informations d'authentification** dans le panneau de navigation.
- Pour ajouter un objet d'informations d'authentification:




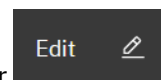
- a) Cliquez sur le bouton Créer  dans la vue de liste des informations d'authentification.
  - b) Indiquez le nom de l'objet des informations d'authentification. Les caractères valides sont les lettres, les chiffres, ainsi que les caractères ".", "/", "\_ " et "%".
  - c) Indiquez le type d'objet d'informations d'authentification.
  - d) Indiquez des informations supplémentaires appropriées pour le type d'objet:
    - Pour **CRL LDAP**, indiquez le **nom du serveur LDAP**. Il s'agit du nom d'hôte, de l'adresse IPv4 à notation décimale à point ou de la notation hexadécimale IPv6 de l'hôte sur lequel s'exécute le serveur LDAP, avec un numéro de port facultatif. Vous pouvez éventuellement spécifier un nom d'utilisateur et un mot de passe pour l'utilisateur qui accède au serveur LDAP.
    - Pour **OCSP**, spécifiez l' **URL du répondeur OCSP**. Il s'agit de l'URL du programme de réponse utilisé pour vérifier la révocation de certificat. Il doit s'agir d'une adresse URL HTTP contenant le nom d'hôte et le numéro de port du canal répondeur OCSP. Si le canal répondeur OCSP utilise le port 80, qui est le port par défaut pour HTTP, le numéro de port peut être omis. Les adresses URL HTTP sont définies dans la norme RFC 1738.
    - Pour **IDPW OS**, aucune exigence supplémentaire n'est requise, bien que vous puissiez éventuellement spécifier d'autres options pour ce type d'authentification.
    - Pour **IDPW LDAP**, indiquez le **nom de serveur LDAP** et le nom d' **utilisateur abrégé** . Le nom du serveur LDAP est le nom d'hôte, IPv4 l'adresse décimale à point ou IPv6 la notation hexadécimale de l'hôte sur lequel s'exécute le serveur LDAP, avec un numéro de port facultatif. Le nom d'utilisateur abrégé est la zone de l'enregistrement utilisateur LDAP qui est utilisée comme nom abrégé pour la connexion. Vous pouvez éventuellement spécifier d'autres options pour ce type d'authentification.
  - e) Cliquez sur **Ajouter**.
- Pour supprimer un objet d'informations d'authentification :

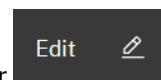


- a) Sélectionnez l'icône représentant une clé  pour l'objet d'informations d'authentification à supprimer de la liste.
  - b) Dans la vue des propriétés de l'objet, cliquez sur **Supprimer l'objet d'informations d'authentification**.
  - c) Confirmez la suppression de l'objet d'informations d'authentification en cliquant sur **Supprimer**. L'objet est supprimé.
- Pour afficher et éditer les propriétés d'un objet d'informations d'authentification :





- a) Sélectionnez l'icône de clé  pour l'objet d'informations d'authentification que vous souhaitez afficher dans la liste.



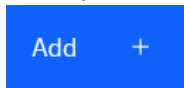
- b) Pour éditer les propriétés affichées, cliquez sur le bouton Editer .
- c) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande.
- d) Cliquez sur **Enregistrer** pour enregistrer vos changements.

- Pour afficher et éditer des enregistrements de droits d'accès pour un objet d'informations d'authentification:

- Sélectionnez l'icône de clé  pour l'objet d'informations d'authentification pour lequel vous souhaitez afficher l'enregistrement de droits d'accès dans la liste.
- Sélectionnez l'onglet **Sécurité**.
- Pour éditer ou supprimer un enregistrement de droits d'accès existant, sélectionnez **Editer** ou

**Supprimer** dans le menu .

- Pour ajouter un nouvel enregistrement de droits d'accès, cliquez sur le bouton **Ajouter**



, indiquez les détails du nouvel enregistrement de droits d'accès et cliquez sur **Créer**.

## **IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente**

Vous pouvez contrôler l'accès des utilisateurs et des groupes aux gestionnaires de files d'attente en spécifiant un enregistrement de droits d'accès pour cet utilisateur ou ce groupe.

### **Pourquoi et quand exécuter cette tâche**

Vous pouvez optimiser l'accès d'un utilisateur de messagerie ou d'un groupe d'utilisateurs de messagerie à un gestionnaire de files d'attente particulier à l'aide d'enregistrements de droits d'accès. Il existe deux types d'enregistrement de droits d'accès: les enregistrements d' **accès au gestionnaire de files d'attente** qui contrôlent les droits généraux et le droit **de créer des enregistrements** qui contrôlent quels utilisateurs et groupes peuvent créer des objets pour le gestionnaire de files d'attente.

### **Procédure**

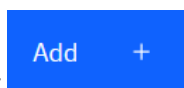
- Pour afficher les enregistrements de droits d'accès d'un gestionnaire de files d'attente:
  - Assurez-vous que le gestionnaire de files d'attente est en cours d'exécution et sélectionnez-le dans la liste des gestionnaires de files d'attente.

- Sélectionnez **Afficher la configuration** dans le menu .

- Vérifiez que l'onglet **Sécurité** est sélectionné.

- Sélectionnez **Enregistrements de droits d'accès** dans le panneau de navigation. La vue affiche les enregistrements de droits d'accès dans deux panneaux, ce qui vous permet de gérer les enregistrements de droits d'accès généraux et de créer des enregistrements de droits d'accès.

- Pour ajouter un enregistrement de droits d'accès généraux, procédez comme suit:



- Cliquez sur le bouton Ajouter  dans la vue de liste **Accès au gestionnaire de files d'attente**.


- Indiquez si vous ajoutez un enregistrement de droits d'accès pour un utilisateur ou un groupe.

- Indiquez le nom de l'utilisateur ou du groupe auquel vous ajoutez un enregistrement de droits d'accès (l'enregistrement de droits d'accès prend ce nom).

- Sélectionnez les droits que vous souhaitez accorder.

- Cliquez sur **Créer**.

- Pour ajouter un enregistrement de création de droits d'accès, procédez comme suit:

- a) Cliquez sur le bouton Ajouter  dans la vue de liste **Droit de création**.
- b) Indiquez si vous ajoutez un enregistrement de droits d'accès pour un utilisateur ou un groupe.
- c) Indiquez le nom de l'utilisateur ou du groupe auquel vous ajoutez un enregistrement de droits d'accès (l'enregistrement de droits d'accès prend ce nom).
- d) Sélectionnez les types d'objet que vous accordez aux droits de création.
- e) Cliquez sur **Créer**.
- Pour supprimer un enregistrement de droits d'accès:
  - a) Sélectionnez l'enregistrement de droits d'accès à supprimer, puis sélectionnez **Supprimer**.
  - b) Confirmez la suppression de l'objet d'informations d'authentification en cliquant sur **Supprimer**. L'objet est supprimé.
- Pour afficher et éditer les propriétés d'un enregistrement de droits d'accès:
  - a) Cliquez sur l'enregistrement de droits d'accès à afficher.
  - b) Modifiez les paramètres selon vos besoins et cliquez sur **Sauvegarder** pour sauvegarder vos modifications.

## **IBM MQ Console: Utilisation des enregistrements d'authentification de canal**


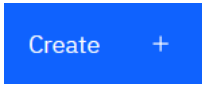
Vous pouvez utiliser IBM MQ Console pour ajouter et supprimer des enregistrements d'authentification de canal sur un gestionnaire de files d'attente. Vous pouvez également afficher et définir les propriétés des enregistrements d'authentification de canal.



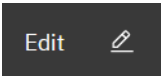
### Pourquoi et quand exécuter cette tâche

Pour exercer un contrôle plus précis sur l'accès accordé aux systèmes se connectant au niveau du canal, vous pouvez utiliser des enregistrements d'authentification de canal.

Pour appliquer la sécurité, vous pouvez utiliser des enregistrements d'authentification de canal bloquants afin de bloquer l'accès à vos canaux. Vous pouvez aussi utiliser des enregistrements d'authentification de canal de mappe d'adresses pour autoriser l'accès à des utilisateurs spécifiés. Pour en savoir plus sur les enregistrements d'authentification de canal, voir [Enregistrements d'authentification de canal](#).

### Procédure

- Pour afficher les informations d'authentification de canal d'un gestionnaire de files d'attente:
  - a) Assurez-vous que le gestionnaire de files d'attente est en cours d'exécution et sélectionnez-le dans la liste des gestionnaires de files d'attente.
  - b) Sélectionnez **Afficher la configuration** dans le menu .
  - c) Vérifiez que l'onglet **Sécurité** est sélectionné.
  - d) Sélectionnez **Authentification de canal** dans le panneau de navigation.
- Pour ajouter un enregistrement d'authentification de canal:
  - a) Cliquez sur le bouton Créer  dans la vue de liste des informations d'authentification de canal.
  - b) Choisissez le type de règle que vous souhaitez utiliser. Sélectionnez **Autoriser**, **Bloquer** ou **Avertir**.
  - c) Choisissez le type d'identité pour lequel vous configurez une règle d'authentification de canal. Différents types d'identité sont disponibles, en fonction du type de règle que vous avez sélectionné.

- d) Indiquez les informations requises pour l'identité que vous spécifiez. Par défaut, les propriétés minimales recommandées sont affichées pour que vous puissiez fournir des valeurs. Vous pouvez afficher toutes les propriétés disponibles en sélectionnant **Création personnalisée**.
- e) Cliquez sur **Créer** pour créer l'enregistrement d'authentification de canal.
- Pour plus d'informations sur les paramètres disponibles pour les enregistrements d'authentification de canal, voir [Enregistrements d'authentification de canal](#) et [SET CHLAUTH](#)
- Pour supprimer un enregistrement d'authentification de canal :
    - a) Cliquez sur l'icône représentant une clé  en regard de l'enregistrement d'authentification de canal à supprimer.
    - b) Dans la vue Editer l'authentification de canal, cliquez sur **Supprimer l'objet d'authentification de canal**.
    - c) Confirmez la suppression de l'enregistrement d'authentification de canal en cliquant sur **Supprimer**. L'enregistrement d'authentification de canal est supprimé.
  - Pour afficher et éditer les propriétés d'un enregistrement d'authentification de canal:
    - a) Cliquez sur l'icône représentant une clé  en regard de l'enregistrement d'authentification de canal que vous souhaitez modifier ou afficher. Les propriétés s'affichent.
    - b) Cliquez sur le bouton Editer 
    - c) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande.
    - d) Cliquez sur **Enregistrer** pour enregistrer vos changements.


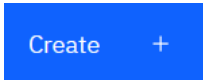
**V 9.4.0** **Multi** **IBM MQ Console: Utilisation des programmes d'écoute**




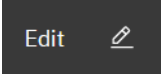
Vous pouvez utiliser le IBM MQ Console pour ajouter et supprimer des programmes d'écoute, démarrer et arrêter des programmes d'écoute, afficher et définir des propriétés de programme d'écoute et gérer les enregistrements de droits d'accès d'un programme d'écoute.



## Pourquoi et quand exécuter cette tâche

La vue Programmes d'écoute affiche les programmes d'écoute qui existent pour un gestionnaire de files d'attente spécifique. Vous pouvez sélectionner des programmes d'écoute individuels à utiliser.

## Procédure

- Pour afficher les programmes d'écoute d'un gestionnaire de files d'attente:
  - a) Assurez-vous que le gestionnaire de files d'attente est en cours d'exécution et sélectionnez-le dans la liste des gestionnaires de files d'attente.
  - b) Sélectionnez **Afficher la configuration** dans le menu .
  - c) Sélectionnez l'onglet **Programmes d'écoute**.
- Pour créer un programme d'écoute:
  - a) Cliquez sur le bouton Créer .
  - b) Fournissez les informations requises pour le programme d'écoute que vous créez.
  - c) Cliquez sur **Créer**. Le nouveau programme d'écoute est créé.
- Pour démarrer un programme d'écoute :

- a) Localisez le programme d'écoute que vous souhaitez démarrer dans la liste.
- b) Sélectionnez **Démarrer** dans le menu .
- Pour arrêter un programme d'écoute :
  - a) Localisez le programme d'écoute que vous souhaitez démarrer dans la liste.
  - b) Sélectionnez **Arrêter** dans le menu .
- Pour afficher et éditer les propriétés d'un programme d'écoute :
  - a) Localisez le programme d'écoute dans la liste.
  - b) Sélectionnez **Afficher la configuration** dans le menu .
  - c) Vérifiez que l'onglet **Propriétés** est sélectionné. Pour éditer les propriétés, cliquez sur le bouton .
 

Editer
  - d) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande. Pour plus d'informations sur les propriétés, voir [Propriétés du programme d'écoute](#) dans la documentation MQ Explorer.
  - e) Cliquez sur **Enregistrer** pour enregistrer vos changements.
- Afin d'afficher et d'éditer des enregistrements de droits d'accès pour un programme d'écoute :
  - a) Localisez le programme d'écoute dans la liste.
  - b) Sélectionnez **Afficher la configuration** dans le menu .
  - c) Cliquez sur l'onglet **Sécurité**.
  - d) Utilisez les enregistrements de droits d'accès décrits pour les enregistrements de droits d'accès du gestionnaire de files d'attente. Voir [«IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente»](#), à la page 106.
- Pour supprimer un programme d'écoute :
  - a) Localisez le programme d'écoute dans la liste.
  - b) Sélectionnez **Afficher la configuration** dans le menu .
  - c) Cliquez sur **Supprimer le programme d'écoute**.

## IBM MQ Console: Ajout d'un gestionnaire de files d'attente éloignées

Vous pouvez utiliser IBM MQ Console pour administrer un gestionnaire de files d'attente exécuté sur un système distant.

### Avant de commencer

- Vous devez préparer le gestionnaire de files d'attente sur le système distant pour qu'il puisse être administré à distance, voir l'étape «1», à la page 112, «2», à la page 112, «3», à la page 112 et «4», à la page 113 de [«Ajout d'un gestionnaire de files d'attente éloignées à IBM MQ Console à l'aide de la ligne de commande»](#), à la page 111.
- Vous devez également activer les connexions distantes à partir de IBM MQ Console. Pour plus d'informations, voir [Configuration du comportement de connexion du gestionnaire de files d'attente éloignées](#).

## Pourquoi et quand exécuter cette tâche

Vous utilisez une table de définition de connexion client (CCDT) au format JSON pour spécifier les détails de la connexion distante. Vous pouvez créer une table de définition de canal du client JSON à l'aide d'un éditeur de texte (voir l'étape «5», à la page 113 de «Ajout d'un gestionnaire de files d'attente éloignées à IBM MQ Console à l'aide de la ligne de commande», à la page 111) ou vous pouvez en créer une à l'aide de IBM MQ Console.

Vous pouvez également créer la table de définition de canal du client à partir de la IBM MQ Console en spécifiant les détails de connexion directement lorsque vous ajoutez le gestionnaire de files d'attente éloignées.

Vous pouvez également connecter un gestionnaire de files d'attente éloignées à IBM MQ Console à l'aide de la ligne de commande pour toutes les tâches requises (en plus de la préparation du gestionnaire de files d'attente éloignées et de la création d'une table de définition de canal du client). Voir «Ajout d'un gestionnaire de files d'attente éloignées à IBM MQ Console à l'aide de la ligne de commande», à la page 111.



**Avertissement :** si vous recevez les messages suivants:

```
MQWB2026E: The request to connect to the remote queue manager 'rqmgr-qmgr_name' failed
with the error message:
'JMSCC0051: The property 'JMS_IBM_MQMD_AccountingToken' should be set using type '[B',
not 'java.lang.Object'.'
```

vous tentez de transmettre un `java.lang.Object` au jeton de comptabilité alors qu'un type d'objet `Java byte[]` est attendu.

## Procédure

- Pour ajouter un gestionnaire de files d'attente éloignées en spécifiant une table de définition de canal du client existante:
  - a) Dans la page d'accueil, cliquez sur **Connecter le gestionnaire de files d'attente éloignées**.
  - b) Nom du gestionnaire de files éloigné.
  - c) Indiquez éventuellement un nom unique pour le gestionnaire de files d'attente. Si vous ne spécifiez pas de nom unique, le nom réel est utilisé avec le préfixe "remote-" ajouté.
  - d) Vérifiez que l'option **Se connecter à l'aide d'une table de définition de canal du client JSON** est sélectionnée.
  - e) Cliquez sur **Parcourir** et sélectionnez le fichier contenant la table de définition de canal du client JSON que vous souhaitez utiliser.
  - f) Cliquez sur **Suivant** pour accéder à la page de l'utilisateur et indiquez éventuellement un nom d'utilisateur et un mot de passe pour vous connecter au gestionnaire de files d'attente éloignées. Si vous ne spécifiez pas ces informations, les informations d'authentification sont extraites du fichier de configuration de la connexion distante.
  - g) Cliquez sur **Suivant** pour accéder à la page Certificat. Si votre table de définition de canal du client spécifie les informations "transmissionSecurity", ces informations sont utilisées. Vous pouvez éventuellement coller un certificat (en tant que clé publique codée en base64) qui est ajouté au magasin de clés de confiance global.

Le certificat est temporairement stocké dans `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` avant d'être ajouté au magasin de clés de confiance. Une fois la connexion ajoutée, le certificat est supprimé de cet emplacement.
  - h) Cliquez sur **Suivant** pour afficher la page récapitulative. Vous pouvez utiliser le bouton **Précédent** pour revisiter les pages précédentes et apporter des corrections. Si vous êtes satisfait des informations, cliquez sur **Connexion** pour vous connecter au gestionnaire de files d'attente éloignées.
- Pour ajouter un gestionnaire de files d'attente éloignées et spécifier manuellement les informations de connexion:

- a) Dans la page d'accueil, cliquez sur **Connecter le gestionnaire de files d'attente éloignées**.
- b) Nom du gestionnaire de files éloigné.
- c) Indiquez éventuellement un nom unique pour le gestionnaire de files d'attente. Si vous ne spécifiez pas de nom unique, le nom réel est utilisé avec le préfixe "remote-" ajouté.
- d) Sélectionnez **Entrée manuelle**.
- e) Entrez le nom du canal de connexion client que la connexion utilisera.
- f) Indiquez le nom de l'hôte sur lequel s'exécute le gestionnaire de files d'attente éloignées. Si des installations MQ distantes sont détectées, les noms d'hôte s'affichent et vous pouvez sélectionner l'hôte du gestionnaire de files d'attente éloignées auquel vous souhaitez vous connecter. Dans certaines configurations réseau, il n'est pas possible de détecter des instances MQ distantes. Dans ce cas, ajoutez le nom d'hôte et le port manuellement.
- g) Cliquez sur **Suivant** pour accéder à la page de l'utilisateur et indiquez éventuellement un nom d'utilisateur et un mot de passe pour vous connecter au gestionnaire de files d'attente éloignées. Si vous ne spécifiez pas ces informations, les informations d'authentification sont extraites du fichier de configuration de la connexion distante.
- h) Cliquez sur **Suivant** pour accéder à la page Certificat. Vous pouvez sélectionner un CipherSpec SSL dans la liste déroulante. Vous pouvez éventuellement coller un certificat (en tant que clé publique codée en base64) qui est ajouté au magasin de clés de confiance global.  
  
Le certificat est temporairement stocké dans `WLP_USER_DIR/generated.certs/uniqueName-qmgrName.crt` avant d'être ajouté au magasin de clés de confiance. Une fois la connexion ajoutée, le certificat est supprimé de cet emplacement.
- i) Cliquez sur **Suivant** pour afficher la page récapitulative. Vous pouvez utiliser le bouton **Précédent** pour revisiter les pages précédentes et apporter des corrections. Si vous êtes satisfait des informations, cliquez sur **Connexion** pour vous connecter au gestionnaire de files d'attente éloignées.

Les informations de connexion que vous avez spécifiées sont écrites dans le fichier CCDT de votre répertoire Web. Le chemin est `WLP_USER_DIR/generated.ccdt/ccdt-uniqueName`.

## Résultats


Le gestionnaire de files d'attente éloignées apparaît dans la liste des gestionnaires de files d'attente éloignées de la IBM MQ Console. A condition que la connexion aboutisse, vous pouvez administrer les objets du gestionnaire de files d'attente éloignées de la même manière que vous utilisez les objets d'un gestionnaire de files d'attente local.

### Ajout d'un gestionnaire de files d'attente éloignées à IBM MQ Console à l'aide de la ligne de commande

Vous pouvez ajouter un gestionnaire de files d'attente éloignées à IBM MQ Console à l'aide de la commande **setmqweb remote** sur la ligne de commande. Un gestionnaire de files d'attente éloignées peut être soit un gestionnaire de files d'attente qui s'exécute dans une installation différente sur le même système que IBM MQ Console, soit un gestionnaire de files d'attente qui s'exécute sur un système différent.

## Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC:

-  Sous AIX, Linux, and Windows, vous émettez des commandes MQSC à partir d'une invite de commande **runmqsc**. Voir [Exécution interactive de commandes MQSC sous runmqsc](#) et [Exécution de commandes MQSC à partir de fichiers texte sous runmqsc](#). Pour cette tâche, si vous utilisez AIX, Linux, and Windows, ouvrez une invite de commande runmqsc qui utilise QM1:

```
runmqsc QM1
```

- **IBM i** Sous IBM i, vous créez une liste de commandes dans un fichier script, puis vous exécutez le fichier à l'aide de la commande **STRMQMMQSC**. Voir [Administration à l'aide de commandes MQSC sous IBM i](#).
- **z/OS** Sous z/OS, les commandes MQSC peuvent être émises à partir de plusieurs sources, en fonction de la commande. Voir [Sources à partir desquelles vous pouvez émettre des commandes MQSC et PCF sous IBM MQ for z/OS](#).

Vérifiez que le serveur mqweb est configuré pour autoriser les connexions de gestionnaire de files d'attente éloignées à IBM MQ Console. Pour plus d'informations, voir [Configuration du comportement des connexions de gestionnaire de files d'attente éloignées](#).

## Procédure

1. Sur le gestionnaire de files d'attente éloignées, créez un canal de connexion serveur pour permettre l'administration à distance du gestionnaire de files d'attente à l'aide de la commande **DEFINE CHANNEL** MQSC.

Par exemple, pour créer un canal de connexion serveur QM1.SVRCONN pour le gestionnaire de files d'attente QM1, entrez la commande MQSC suivante:

```
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
```

Pour plus d'informations sur **DEFINE CHANNEL** et les options disponibles, voir [DEFINE CHANNEL](#).

2. Vérifiez qu'un utilisateur approprié est autorisé à administrer le gestionnaire de files d'attente et les objets MQ associés au gestionnaire de files d'attente.

- **ALW** Sous AIX, Linux, and Windows, utilisez la commande de contrôle **setmqaut** sur une ligne de commande standard.
- **z/OS** Sous z/OS, définissez des profils RACF pour accorder à l'utilisateur autorisé l'accès au gestionnaire de files d'attente.

Par exemple, sous AIX, Linux, and Windows, pour autoriser l'utilisateur `exampleUser` à accéder au gestionnaire de files d'attente QM1, entrez la commande de contrôle suivante:

```
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

Cet utilisateur autorisé peut être l'un des utilisateurs suivants:

- ID utilisateur identique à l'ID utilisateur qui démarre le serveur mqweb qui exécute IBM MQ Console sur le système à partir duquel vous souhaitez administrer à distance ce gestionnaire de files d'attente.
- ID utilisateur correspondant à un ID utilisateur et à un mot de passe qui sont ensuite inclus dans la commande **setmqweb remote** à l'étape «7», à la page 113. En incluant l'ID utilisateur et le mot de passe dans la commande **setmqweb remote**, cet ID utilisateur et ce mot de passe sont utilisés pour l'authentification lorsque IBM MQ Console se connecte au gestionnaire de files d'attente.
- ID utilisateur déterminé par les règles de sécurité du canal. Par exemple, vous pouvez définir une règle d'authentification de canal sur le canal de connexion serveur pour autoriser les connexions à partir de l'adresse IP à partir de laquelle vous utilisez IBM MQ Console pour l'administration à distance, et mapper toutes ces connexions à un ID utilisateur spécifique autorisé à utiliser le gestionnaire de files d'attente. Pour plus d'informations, voir [Création de nouvelles règles CHLAUTH pour les canaux](#).

3. **ALW**


Si aucun programme d'écoute n'est en cours d'exécution sur le gestionnaire de files d'attente éloignées, créez un programme d'écoute pour accepter les connexions réseau entrantes à l'aide de la commande **DEFINE LISTENER** MQSC.




Par exemple, pour créer un programme d'écoute REMOTE . LISTENER sur le port 1414 pour le gestionnaire de files d'attente éloignées QM1, entrez la commande MQSC suivante:

```
runmqsc QM1
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
end
```

4. Vérifiez que le programme d'écoute est en cours d'exécution à l'aide de la commande **START LISTENER** MQSC.

 Par exemple, sous AIX, Linux, and Windows , pour démarrer le programme d'écoute REMOTE . LISTENER pour le gestionnaire de files d'attente QM1, entrez la commande MQSC suivante:

```
runmqsc QM1
START LISTENER(REMOTE.LISTENER)
end
```

 Par exemple, sous z/OS, pour démarrer le programme d'écoute, entrez la commande MQSC suivante:

```
/cpcf START LISTENER TRPTYPE(TCP) PORT(1414)
```

Notez que l'espace adresse de l'initiateur de canal doit être démarré pour que vous puissiez démarrer un programme d'écoute sur z/OS.

5. Créez un fichier CCDT JSON contenant les informations de connexion du gestionnaire de files d'attente éloignées:

- Générez un fichier CCDT à l'aide du IBM MQ Console associé à la même installation que le gestionnaire de files d'attente auquel vous souhaitez vous connecter à distance.

Dans le panneau **Accueil** , cliquez sur la vignette **Télécharger le fichier de connexion** .

- Créez un fichier CCDT au format JSON qui définit la connexion. Pour plus d'informations sur la création d'une table de définition de canal du client au format JSON, voir [Configuration d'une table de définition de canal du client au format JSON](#).

Le fichier CCDT doit inclure les informations `name`, `clientConnection` et `type` . Vous pouvez éventuellement inclure des informations supplémentaires, telles que des informations `transmissionSecurity` . Pour plus d'informations sur toutes les définitions d'attribut de canal CCDT, voir [Liste complète des définitions d'attribut de canal CCDT](#).

L'exemple suivant illustre un fichier CCDT JSON de base pour une connexion de gestionnaire de files d'attente éloignées. Il définit le nom du canal sur le même nom que l'exemple de canal de connexion serveur créé à l'étape «1», à la page 112, et le port de connexion sur la même valeur que le port utilisé par le programme d'écoute. L'hôte de connexion est défini sur le nom d'hôte du système sur lequel l'exemple de gestionnaire de files d'attente éloignées, QM1, est en cours d'exécution:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

6. Copiez le fichier CCDT JSON sur le système sur lequel IBM MQ Console s'exécute.
7. A partir de l'installation qui exécute IBM MQ Console, utilisez la commande **setmqweb remote** pour ajouter les informations du gestionnaire de files d'attente éloignées à la configuration IBM MQ Console .

Au minimum, pour ajouter un gestionnaire de files d'attente éloignées à IBM MQ Console, vous devez fournir le nom du gestionnaire de files d'attente, un nom unique pour le gestionnaire de files d'attente (pour différencier les autres gestionnaires de files d'attente éloignées qui peuvent avoir le même nom de gestionnaire de files d'attente) et l'URL CCDT du gestionnaire de files d'attente. Le nom unique est le nom d'affichage dans IBM MQ Console. Par conséquent, spécifiez un nom qui indique clairement qu'il s'agit d'un gestionnaire de files d'attente éloigné, par exemple, "remote-QM2". Vous pouvez spécifier plusieurs options supplémentaires, telles que le nom d'utilisateur et le mot de passe à utiliser pour la connexion du gestionnaire de files d'attente éloignées ou les détails du magasin de clés de confiance et du magasin de clés. Pour obtenir la liste complète des paramètres pouvant être spécifiés avec la commande **setmqweb remote**, voir [setmqweb remote](#).

Par exemple, pour ajouter l'exemple de gestionnaire de files d'attente éloignées QM1, à l'aide de l'exemple de fichier CCDT, entrez la commande suivante:

```
setmqweb remote add -uniqueName "MACHINEAQM1" -qmgrName "QM1" -ccdtURL "c:\myccdts\ccdt.json"
```

## Résultats

Le gestionnaire de files d'attente éloignées apparaît dans la liste des gestionnaires de files d'attente éloignées dans IBM MQ Console lorsque la liste des connexions éloignées est régénérée. A condition que la connexion aboutisse, vous pouvez administrer les objets du gestionnaire de files d'attente éloignées de la même manière que vous utilisez les objets d'un gestionnaire de files d'attente local.

## Exemple

L'exemple suivant configure la connexion de gestionnaire de files d'attente éloignées pour un gestionnaire de files d'attente QM1. IBM MQ Console est autorisé à administrer le gestionnaire de files d'attente en fonction de l'autorisation accordée à l'utilisateur `exampleUser`. Les données d'identification de cet utilisateur sont fournies à IBM MQ Console lorsque la commande **setmqweb remote** est utilisée pour configurer les informations de connexion du gestionnaire de files d'attente éloignées.

1. Sur le système où se trouve le gestionnaire de files d'attente éloignées QM1, un canal de connexion serveur et un programme d'écoute sont créés. Le programme d'écoute est démarré et l'utilisateur `exampleUser` est autorisé à administrer le gestionnaire de files d'attente. Par exemple, sous AIX, Linux, and Windows, exécutez les commandes suivantes:

```
runmqsc QM1
#Define the server connection channel that will accept connections from the Console
DEFINE CHANNEL(QM1.SVRCONN) CHLTYPE(SVRCONN) TRPTYPE(TCP)
# Define the listener to use for the connection from the Console
DEFINE LISTENER(REMOTE.LISTENER) TRPTYPE(TCP) PORT(1414)
# Start the listener
START LISTENER(REMOTE.LISTENER)
end

#Set mq authorization for exampleUser to access the queue manager
setmqaut -m QM1 -t qmgr -p exampleUser +connect +inq +setall +dsp
```

2. Sur le système où le IBM MQ Console est en cours d'exécution, un fichier `QM1_ccdt.json` est créé avec les informations de connexion suivantes:

```
{
  "channel": [{
    "name": "QM1.SVRCONN",
    "clientConnection": {
      "connection": [{
        "host": "example.com",
        "port": 1414
      }],
      "queueManager": "QM1"
    },
    "type": "clientConnection"
  }]
}
```

3. Sur le système sur lequel IBM MQ Console s'exécute, les informations de connexion du gestionnaire de files d'attente éloignées pour le gestionnaire de files d'attente QM1 sont ajoutées au serveur mqweb. Les données d'identification de `exampleUser` sont incluses dans les informations de connexion:

```
setmqweb remote add -uniqueName "remote-QM1" -qmgrName "QM1" -ccdtURL  
"c:\myccdt\QM1_ccdt.json" -username "exampleUser" -password "password"
```

4. La IBM MQ Console montre le gestionnaire de files d'attente éloignées QM1.

## **V 9.4.0 IBM MQ Console: Gestion des objets**

Chaque gestionnaire de files d'attente IBM MQ est associé à plusieurs types d'objet différents.

### **Pourquoi et quand exécuter cette tâche**

Vous pouvez utiliser la console pour utiliser les types d'objet IBM MQ suivants:

- Files d'attente
- Objets d'événements:
  - Rubriques
  - Abonnements
- Objets d'application:
  - Connexions
  - Canaux d'application
  - Instances de canal d'application
- Objets réseau MQ :
  - Gestionnaires de files d'attente connectés
  - Canaux de gestionnaire de file d'attente
  - Instances de canal de gestionnaire de files d'attente

### **Procédure**

Pour utiliser un objet IBM MQ :


1. Dans la vue de liste des gestionnaires de files d'attente, cliquez sur le gestionnaire de files d'attente qui possède les objets que vous souhaitez utiliser.
2. Cliquez sur l'onglet de réseau Files d'attente, Événements, Applications ou MQ pour sélectionner le type d'objet à utiliser.
3. Consultez l'une des rubriques suivantes pour obtenir des instructions détaillées sur l'utilisation des objets.

## **V 9.4.0 IBM MQ Console: Utilisation des files d'attente**

Vous pouvez afficher les files d'attente qui existent pour un gestionnaire de files d'attente spécifique dans l'onglet **Files d'attente** . Vous pouvez ajouter et supprimer des files d'attente, ajouter et supprimer des messages dans une file d'attente, parcourir des messages, afficher et définir les propriétés d'une file d'attente et gérer les enregistrements de droits d'accès d'une file d'attente.

### **Pourquoi et quand exécuter cette tâche**

La vue des files d'attente répertorie les files d'attente qui existent pour un gestionnaire de files d'attente spécifique. Vous pouvez accéder à la liste des files d'attente en cliquant sur un gestionnaire de files d'attente et en sélectionnant l'onglet **Files d'attente** . Vous pouvez sélectionner des files d'attente individuelles dans la liste à gérer.

 Vous ne pouvez pas afficher ni éditer les enregistrements de droits d'accès pour les files d'attente sur z/OS.

## Procédure

- Pour ajouter une file d'attente :

a) Dans l'onglet **Files d'attente** , cliquez sur le bouton Créer



b) Sélectionnez le type de file d'attente à créer:

- File d'attente locale-stockez les messages dans le gestionnaire de files d'attente auquel il appartient.
- File d'attente alias-un pointeur vers une autre file d'attente sur le même gestionnaire de files d'attente.
- File d'attente éloignée-un pointeur vers une autre file d'attente sur un autre gestionnaire de files d'attente.
- File d'attente modèle: modèle pour une file d'attente utilisée lors de la création d'un gestionnaire de files d'attente dynamique.

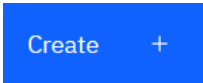
c) Indiquez les informations requises pour le type de file d'attente que vous créez. Par défaut, les propriétés minimales recommandées sont affichées pour que vous puissiez fournir des valeurs. Vous pouvez afficher toutes les propriétés disponibles en sélectionnant **Création personnalisée**.

d) Cliquez sur **Créer**. La nouvelle file d'attente est créée.

- Pour placer des messages dans une file d'attente:

a) Cliquez sur la file d'attente à laquelle vous souhaitez ajouter des messages dans la liste de la vue de liste des files d'attente. Vous ne pouvez pas sélectionner de file d'attente modèle.

b) Cliquez sur le bouton Créer



c) Entrez le message que vous souhaitez placer dans la file d'attente.

d) Cliquez sur **Créer**.

- Pour effacer des messages dans une file d'attente :

a) Cliquez sur la file d'attente locale dont vous souhaitez supprimer les messages dans la liste des files d'attente.


b) Cliquez sur l'icône Effacer la file d'attente 

c) Confirmez l'effacement de la file d'attente en cliquant sur **Effacer la file d'attente**.

- 

Pour supprimer un message individuel d'une file d'attente:

a) Recherchez le message à supprimer.

b) Cliquez sur l'icône de suppression en regard du message 

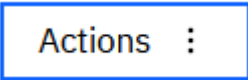

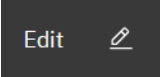

c) Confirmez que vous souhaitez effacer le message en cliquant sur **Supprimer**.

- Pour parcourir les messages d'une file d'attente, cliquez sur la file d'attente dans la vue de liste des files d'attente. La liste des messages de cette file d'attente s'affiche.

- Pour supprimer une file d'attente :


a) Cliquez sur la file d'attente locale à supprimer dans la liste des files d'attente.

Actions :

- b) Cliquez sur le bouton Actions  et sélectionnez **Supprimer la file d'attente**.
- c) Confirmez la suppression de la file d'attente en cliquant sur **Supprimer**. La file d'attente est supprimée.
- Pour afficher et éditer les propriétés d'une file d'attente :
  - a) Sélectionnez **Afficher la configuration** dans le menu  en regard de la file d'attente à éditer.
  - b) Cliquez sur le bouton Editer 
  - c) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande. Pour plus d'informations sur les propriétés, voir [Propriétés de file d'attente](#) dans la documentation IBM MQ Explorer
  - d) Cliquez sur **Enregistrer** pour enregistrer vos changements.
- Afin d'afficher et d'éditer des enregistrements de droits d'accès pour une file d'attente :
  - a) Sélectionnez **Afficher la configuration** dans le menu  en regard de la file d'attente à éditer.
  - b) Cliquez sur l'onglet **Sécurité**.
  - c) Utilisez les enregistrements de droits d'accès décrits pour les enregistrements de droits d'accès du gestionnaire de files d'attente. Voir «IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente», à la page 106.

V 9.4.0

Pour afficher les objets IBM MQ associés à une file d'attente:

- a) Sélectionnez **Afficher les objets associés** dans le menu  en regard de la file d'attente à afficher.
- b) Affichez les objets dans le panneau qui s'affiche. Cliquez sur les liens pour afficher plus de détails sur chacun des objets répertoriés.

Vous pouvez utiliser ce panneau pour afficher les applications qui placent des messages dans des files d'attente et voir les relations entre les différentes files d'attente. Cela peut vous aider à identifier et à résoudre les problèmes.

V 9.4.0

### **IBM MQ Console: Utilisation des rubriques**

Vous pouvez utiliser le IBM MQ Console pour ajouter et supprimer des rubriques, ainsi que pour afficher et définir les propriétés d'une rubrique.

## **Pourquoi et quand exécuter cette tâche**

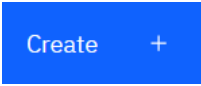



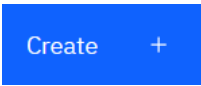
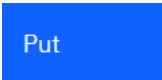

La vue des rubriques répertorie les rubriques qui existent pour un gestionnaire de files d'attente spécifique. Vous pouvez accéder aux rubriques à partir de l'onglet **Evénements** du gestionnaire de files d'attente. Vous pouvez sélectionner des rubriques individuelles dans la liste à utiliser.

z/OS

Vous ne pouvez pas afficher ou éditer les enregistrements de droits d'accès pour une rubrique dans z/OS.

## **Procédure**

- Pour ajouter une rubrique :
  - a) Dans la vue du gestionnaire de files d'attente, ouvrez l'onglet **Evénements** et cliquez sur **Rubriques**.


- b) Cliquez sur le bouton Créer .
- c) Indiquez les informations requises pour la rubrique que vous créez. Par défaut, les propriétés minimales recommandées sont affichées pour que vous puissiez fournir des valeurs. Vous pouvez afficher toutes les propriétés disponibles en sélectionnant **Création personnalisée**.
- d) Cliquez sur **Créer**. La nouvelle rubrique est créée.
- Pour supprimer une rubrique :
    - a) Cliquez sur l'icône représentant une clé  en regard de la rubrique à supprimer.
    - b) Dans la vue Editer la file d'attente, cliquez sur **Supprimer la rubrique**.
    - c) Confirmez la suppression de la rubrique en cliquant sur **Supprimer**. La rubrique est supprimée.
  - Pour afficher et éditer les propriétés d'une rubrique :
    - a) Cliquez sur l'icône représentant une clé  en regard de la rubrique à éditer.
    - b) Cliquez sur le bouton Editer .
    - c) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande. Pour plus d'informations sur les propriétés, voir [Propriétés de rubrique](#) dans la documentation MQ Explorer.
    - d) Cliquez sur **Enregistrer** pour enregistrer vos changements.
  - Pour publier un message sur une rubrique, vous devez disposer d'au moins un abonnement correspondant.
    - a) Cliquez sur la rubrique dans laquelle vous souhaitez effectuer la publication dans la liste des rubriques.
    - b) Cliquez sur le nom de l'abonnement correspondant.
    - c) Cliquez sur le bouton Créer .
    - d) Entrez le message à publier.
    - e) Cliquez sur le bouton Insérer . Le message est écrit dans tous les abonnements correspondants.
  - Pour vous abonner à une rubrique, voir [«IBM MQ Console: Utilisation des abonnements»](#), à la page 119:
  - Afin d'afficher et d'éditer des enregistrements de droits d'accès pour une rubrique :
    - a) Cliquez sur l'icône de clé  en regard de la rubrique pour laquelle vous souhaitez éditer les enregistrements de droits d'accès.
    - b) Cliquez sur l'onglet **Sécurité**.
    - c) Utilisez les enregistrements de droits d'accès comme décrit pour les enregistrements de droits d'accès de gestionnaire de files d'attente. Voir [«IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente»](#), à la page 106.

Vous pouvez utiliser le IBM MQ Console pour ajouter et supprimer des abonnements, et afficher et définir les propriétés d'un abonnement.

## Pourquoi et quand exécuter cette tâche

La vue des abonnements répertorie les abonnements qui existent pour un gestionnaire de files d'attente spécifique. Vous pouvez accéder aux abonnements à partir de l'onglet **Événements** du gestionnaire de files d'attente. Vous pouvez sélectionner des rubriques individuelles dans la liste à utiliser. Vous pouvez sélectionner des abonnements individuels dans la liste à utiliser.


Pour plus d'informations sur les abonnements, voir [Abonnés et abonnements](#) et [DEFINE SUB](#).

 Vous ne pouvez pas afficher ni éditer les enregistrements de droits d'accès pour un abonnement sur z/OS.

## Procédure

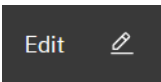
- Pour ajouter un abonnement:
  - a) Dans la vue du gestionnaire de files d'attente, ouvrez l'onglet **Événements** et cliquez sur **Abonnements**.
  - b) Indiquez si vous souhaitez créer un abonnement géré ou non géré.
  - c) Indiquez les informations requises pour l'abonnement que vous créez. Par défaut, les propriétés minimales recommandées sont affichées pour que vous puissiez fournir des valeurs. Vous pouvez afficher toutes les propriétés disponibles en sélectionnant **Création personnalisée**.
  - d) Cliquez sur **Créer**. Le nouvel abonnement est créé.

- Pour supprimer un abonnement:

- a) Cliquez sur l'icône représentant une clé  en regard de l'abonnement à supprimer.
- b) Dans la vue Editer la file d'attente, cliquez sur **Supprimer l'abonnement**.
- c) Confirmez que vous souhaitez supprimer l'abonnement en cliquant sur **Supprimer**. L'abonnement est supprimé.

- Pour afficher et éditer les propriétés d'un abonnement:

- a) Cliquez sur l'icône représentant une clé  en regard de l'abonnement à éditer.

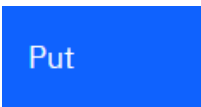
- b) Cliquez sur le bouton Editer .
- c) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande.
- d) Cliquez sur **Enregistrer** pour enregistrer vos changements.

- Pour publier un message sur la rubrique à laquelle l'abonnement est abonné:

- a) Cliquez sur l'abonnement dont vous souhaitez publier la rubrique dans la liste des abonnements.

- b) Cliquez sur le bouton Créer .

- c) Entrez le message à publier.


- d) Cliquez sur le bouton Insérer . Le message est écrit dans tous les abonnements qui correspondent à la rubrique dans laquelle vous avez publié.

## **IBM MQ Console: Utilisation des canaux de gestionnaire de files d'attente**

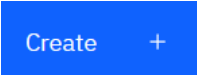




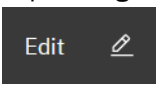
Vous pouvez utiliser IBM MQ Console avec des canaux de gestionnaire de files d'attente: vous pouvez ajouter et supprimer des canaux de gestionnaire de files d'attente, démarrer et arrêter des canaux, réinitialiser et résoudre des canaux et des canaux ping. Vous pouvez également afficher et définir les propriétés d'un canal de gestionnaire de files d'attente et gérer les enregistrements de droits d'accès pour le canal.

### **Pourquoi et quand exécuter cette tâche**






Un canal de gestionnaire de files d'attente est une liaison de communication logique pour la transmission de messages entre les gestionnaires de files d'attente sur un réseau. La vue de canal du gestionnaire de files d'attente comprend un panneau qui affiche une vue rapide du nombre de canaux en cours d'exécution, du nombre de nouvelles tentatives et du nombre de canaux arrêtés.

 Vous ne pouvez pas afficher ni éditer les enregistrements de droits d'accès d'un canal sur z/OS.

### **Procédure**

- Pour ajouter un canal de gestionnaire de files d'attente:
  - a) Dans la vue du gestionnaire de files d'attente, ouvrez l'onglet **MQ network** et cliquez sur **Canaux de gestionnaire de files d'attente** et cliquez sur le bouton **Créer** .
  - b) Sélectionnez le type de canal de gestionnaire de files d'attente que vous souhaitez créer et cliquez sur le bouton suivant .
  - c) Fournissez les informations requises pour le canal que vous créez. Par défaut, les propriétés minimales recommandées sont affichées pour que vous puissiez fournir des valeurs. Vous pouvez afficher toutes les propriétés disponibles en sélectionnant **Création personnalisée**.
  - d) Cliquez sur **Créer**. Le nouveau canal est créé avec le statut **inactif**.
- Pour démarrer un canal de gestionnaire de files d'attente:
  - a) Localisez le canal que vous souhaitez démarrer dans la liste.
  - b) Sélectionnez **Démarrer** dans le menu .
- Pour arrêter un canal de gestionnaire de files d'attente:
  - a) Recherchez le canal que vous souhaitez arrêter dans la liste.
  - b) Sélectionnez **Arrêter** dans le menu .
- Pour afficher les propriétés d'un canal de gestionnaire de files d'attente:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Afficher les configurations** dans le menu .
  - c) Vérifiez que l'onglet **Propriétés** est sélectionné. Pour éditer les propriétés, cliquez sur le bouton **Editer** .
  - d) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande. Pour plus d'informations sur les propriétés, voir [Propriétés du canal](#) dans la documentation MQ Explorer.




- e) Cliquez sur **Enregistrer** pour enregistrer vos changements.
- Pour réinitialiser un canal de gestionnaire de files d'attente:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Avancé** dans le menu  .
  - c) Dans la section **Réinitialiser** , indiquez un numéro de séquence de message.  
Vous devez réinitialiser un canal s'il ne démarre pas car les deux extrémités divergent sur le numéro de message du message suivant à envoyer. Le numéro de séquence de message spécifie ce numéro.
  - d) Cliquez sur **Réinitialiser le canal**.
- Pour résoudre un canal émetteur ou serveur:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Avancé** dans le menu  .
  - c) Dans la section **Résoudre** , choisissez de valider ou d'annuler le lot de messages en cours en cliquant sur **Restaurer les messages dans la file d'attente de transmission** ou sur **Supprimer les messages**.
- Pour envoyer une commande ping à un canal de gestionnaire de files d'attente:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Ping** dans le menu  .
- Pour afficher et éditer des enregistrements de droits d'accès pour un canal de gestionnaire de files d'attente:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Afficher la configuration** dans le menu  .
  - c) Cliquez sur l'onglet **Sécurité**.
  - d) Utilisez les enregistrements de droits d'accès comme décrit pour les enregistrements de droits d'accès de gestionnaire de files d'attente. Voir «[IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente](#)», à la page 106.
- Pour supprimer un canal de gestionnaire de files d'attente:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Configurer** dans le menu  .
  - c) Cliquez sur **Supprimer le canal**.

#### **V 9.4.0 IBM MQ Console: Utilisation des canaux d'application**

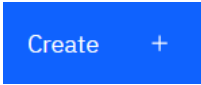




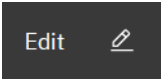

Vous pouvez utiliser le IBM MQ Console pour travailler avec des canaux d'application: vous pouvez ajouter et supprimer des canaux, démarrer et arrêter des canaux, réinitialiser et résoudre des canaux et des canaux ping. Vous pouvez également afficher et définir les propriétés d'un canal d'application et gérer les enregistrements de droits d'accès pour le canal.





### **Pourquoi et quand exécuter cette tâche**

Un canal d'application est une liaison de communication logique utilisée par les applications pour se connecter à un gestionnaire de files d'attente sur un réseau. La vue du canal d'application comprend un panneau qui affiche une vue rapide du nombre de canaux en cours d'exécution, du nombre de nouvelles tentatives et du nombre de canaux arrêtés.

 Vous ne pouvez pas afficher ni éditer les enregistrements de droits d'accès d'un canal sur z/OS.

## Procédure

- Pour ajouter un canal d'application:
  - a) Dans la vue du gestionnaire de files d'attente, ouvrez l'onglet **Applications**, cliquez sur **Canaux d'application** et cliquez sur le bouton Créer .
  - b) Cliquez sur le bouton suivant .
  - c) Fournissez les informations requises pour le canal que vous créez. Par défaut, les propriétés minimales recommandées sont affichées pour que vous puissiez fournir des valeurs. Vous pouvez afficher toutes les propriétés disponibles en sélectionnant **Création personnalisée**.
  - d) Cliquez sur **Créer**. Le nouveau canal est créé avec le statut **inactif**.
- Pour démarrer un canal d'application:
  - a) Localisez le canal que vous souhaitez démarrer dans la liste.
  - b) Sélectionnez **Démarrer** dans le menu .
- Pour arrêter un canal d'application:
  - a) Recherchez le canal que vous souhaitez arrêter dans la liste.
  - b) Sélectionnez **Arrêter** dans le menu .
- Pour afficher les propriétés d'un canal d'application:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Afficher la configuration** dans le menu .
  - c) Vérifiez que l'onglet **Propriétés** est sélectionné. Pour éditer les propriétés, cliquez sur le bouton  Editer.
  - d) Editez les propriétés de façon appropriée. Si la zone de texte de la propriété est désactivée, la propriété est en lecture seule ou peut être éditée uniquement à partir de la ligne de commande. Pour plus d'informations sur les propriétés, voir [Propriétés du canal](#) dans la documentation MQ Explorer.
  - e) Cliquez sur **Enregistrer** pour enregistrer vos changements.
- Pour réinitialiser un canal d'application:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Avancé** dans le menu .
  - c) Dans la section **Réinitialiser**, indiquez un numéro de séquence de message. Vous devez réinitialiser un canal s'il ne démarre pas car les deux extrémités divergent sur le numéro de message du message suivant à envoyer. Le numéro de séquence de message spécifie ce numéro.
  - d) Cliquez sur **Réinitialiser le canal**.
- Pour résoudre un canal émetteur ou serveur:
  - a) Localisez le canal dans la liste.

- b) Sélectionnez **Avancé** dans le menu  .
  - c) Dans la section **Résoudre** , choisissez de valider ou d'annuler le lot de messages en cours en cliquant sur **Restaurer les messages dans la file d'attente de transmission** ou sur **Supprimer les messages**.
- Pour effectuer le test Ping d'un canal, procédez comme suit :
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Ping** dans le menu  .
- Pour afficher et éditer des enregistrements de droits d'accès pour un canal d'application:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Configurer** dans le menu  .
    - c) Cliquez sur l'onglet **Sécurité**.
    - d) Utilisez les enregistrements de droits d'accès comme décrit pour les enregistrements de droits d'accès de gestionnaire de files d'attente. Voir [«IBM MQ Console: Utilisation des enregistrements de droits d'accès du gestionnaire de files d'attente»](#), à la page 106 .
- Pour supprimer un canal d'application:
  - a) Localisez le canal dans la liste.
  - b) Sélectionnez **Configurer** dans le menu  .
    - c) Cliquez sur **Supprimer le canal**.

### **IBM MQ Console: Utilisation des applications**

Vous pouvez utiliser l' IBM MQ Console pour afficher des informations sur les applications connectées à un gestionnaire de files d'attente.

## Pourquoi et quand exécuter cette tâche

Une application est connectée à un gestionnaire de files d'attente via un réseau à l'aide d'un canal de connexion serveur. La vue des applications comprend un panneau qui affiche une vue rapide du nombre d'applications connectées à un gestionnaire de files d'attente.

## Procédure

- Pour afficher les informations d'application:
  - a) Dans la vue du gestionnaire de files d'attente, ouvrez l'onglet **Applications** .
  - b) Cliquez sur **Applications connectées** pour ouvrir la vue des applications.
  - c) S'il existe plusieurs instances d'une application, cliquez sur la flèche vers le bas pour afficher les détails de chaque instance.
  - d) Cliquez sur des objets dans la vue pour obtenir plus de détails.

### **IBM MQ Console: Working with storage classes**

You can use the IBM MQ Console to add, view, delete and update storage classes on z/OS queue managers.

## About this task

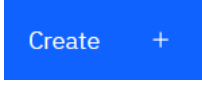
The storage classes view lists the storage classes that exist for a specific queue manager. You access **Storage classes** from the sidebar on the queue manager **Queues** tab.

See [Storage classes for IBM MQ for z/OS](#) and [DEFINE STGLASS](#) for more information about storage classes.

## Procedure

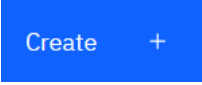
- To add a storage class:

a) From the queue manager view, open the **Queues** tab, and click **Storage classes**.

b) On the **Storage classes** screen, click the **Create**  button.

c) Provide the required information for the storage class you are creating.

By default, the minimum recommended properties you need to provide values for are displayed. You can view all of the available properties by selecting **Custom create**.

d) Click the **Create**  button.

The new storage class is created.

- To delete a storage class:



a) Click the spanner button  next to the storage class that you want to delete.

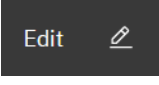
b) In the Edit storage class view, click **Delete storage class**.

c) Confirm that you want to delete the queue by clicking **Delete**. The storage class is deleted.

- To view and edit the properties of a storage class:



a) Click the spanner button  next to the storage class that you want to edit.

b) Click the Edit button .

c) Edit the properties as required. If the property text box is disabled, the property is read-only, or can be set only at the time of creation.

d) Click **Save** to save your changes.

## **IBM MQ Console: Working with page sets and buffer pools**

You can use the IBM MQ Console to view page sets and buffer pools on z/OS queue managers.

## About this task

The page sets and buffer pools views list the page sets and buffer pools that exist for a specific queue manager. You access the **Page sets** and **Buffer pools** views from the sidebar of the queue manager **Queues** tab

See [Page sets for IBM MQ for z/OS](#) for more information about page sets, and [Buffers and buffer pools for IBM MQ for z/OS](#) for more information about buffer pools.

## Procedure

- To view the properties of a page set



Click the spanner button next to the page set that you want to view.


- To view the properties of a buffer pool



Click the spanner button next to the buffer pool that you want to view.

## Paramètres IBM MQ Console

Vous pouvez spécifier des paramètres généraux pour le IBM MQ Console.

Cliquez sur l'icône des paramètres  pour passer à la vue des paramètres IBM MQ Console .

Utilisez les paramètres pour contrôler les fonctions suivantes:

- Régénération automatique des gestionnaires de files d'attente toutes les dix secondes. Cette fonction peut être activée ou désactivée.
- Indique si les objets système sont affichés. Vous pouvez le spécifier pour tous les types d'objet ou sélectionner des types d'objet individuellement.
- Indique si les informations de trace sont collectées ou non.

Windows

Linux

## Administration à l'aide de IBM MQ Explorer

IBM MQ Explorer vous permet d'effectuer l'administration locale ou à distance de votre réseau à partir d'un ordinateur exécutant Windows, ou Linux x86-64 uniquement.

IBM MQ for Windows et IBM MQ for Linux x86-64 fournissent une interface d'administration appelée IBM MQ Explorer pour effectuer des tâches d'administration au lieu d'utiliser des commandes de contrôle ou MQSC. [La comparaison des ensembles de commandes](#) vous montre ce que vous pouvez faire à l'aide de l'IBM MQ Explorer.

Le IBM MQ Explorer vous permet d'effectuer une administration locale ou à distance de votre réseau à partir d'un ordinateur exécutant Windows, ou Linux x86-64, en pointant le IBM MQ Explorer vers les gestionnaires de files d'attente et les clusters qui vous intéressent. Il peut se connecter à distance aux gestionnaires de files d'attente s'exécutant sur n'importe quelle plateforme prise en charge, notamment z/OS, ce qui vous permet de visualiser, d'explorer et de modifier l'intégralité de votre réseau principal de messagerie depuis la console.

Pour configurer les gestionnaires de files d'attente IBM MQ distantes afin que IBM MQ Explorer puisse les administrer, voir [«Logiciels prérequis et définitions pour IBM MQ Explorer»](#), à la page 127.

Il vous permet d'effectuer des tâches, généralement associées à la configuration et à l'optimisation de l'environnement de travail pour IBM MQ, localement ou à distance dans un domaine système Windows ou Linux x86-64 .

Sous Linux, le démarrage de IBM MQ Explorer peut échouer si vous disposez de plusieurs installations Eclipse . Si cela se produit, démarrez IBM MQ Explorer à l'aide d'un ID utilisateur différent de celui que vous utilisez pour l'autre installation Eclipse .

Sous Linux, pour démarrer correctement IBM MQ Explorer , vous devez pouvoir écrire un fichier dans votre répertoire de base et ce dernier doit exister.

IBM MQ Explorer peut être installé à partir du téléchargement IBM MQ Explorer autonome disponible à partir de Fix Central. Pour plus d'informations, voir [Installation et désinstallation de IBM MQ Explorer en tant qu'application autonome sous Linux et Windows](#).

## Ce que vous pouvez faire avec le IBM MQ Explorer

Vous pouvez utiliser IBM MQ Explorer pour effectuer des tâches d'administration à l'aide d'une série de vues de contenu et de boîtes de dialogue de propriétés. Vous pouvez également étendre le IBM MQ Explorer en écrivant un ou plusieurs plug-in Eclipse .

### Tâches IBM MQ Explorer

Avec IBM MQ Explorer, vous pouvez effectuer les tâches suivantes:

- Créez et supprimez un gestionnaire de files d'attente (sur votre machine locale uniquement).
- Démarrez et arrêtez un gestionnaire de files d'attente (sur votre machine locale uniquement).
- Définissez, affichez et modifiez les définitions des objets IBM MQ tels que les files d'attente et les canaux.
- Parcourir les messages d'une file d'attente.
- Démarrage et arrêt d'un canal.
- Affichez des informations de statut sur un canal, un programme d'écoute, une file d'attente ou des objets de service.
- Afficher les gestionnaires de files d'attente dans un cluster.
- Cochez cette case pour voir quelles applications, quels utilisateurs ou quels canaux ont une file d'attente particulière ouverte.
- Créez un cluster de gestionnaires de files d'attente à l'aide de l'assistant de création de cluster.
- Ajout d'un gestionnaire de files d'attente à un cluster à l'aide de l'assistant Ajout d'un gestionnaire de files d'attente à un cluster.
- Gérez l'objet d'informations d'authentification, utilisé avec la sécurité de canal TLS (Transport Layer Security).
- Créer et supprimer des initiateurs de canal, des moniteurs de déclenchement et des programmes d'écoute.
- Démarrez ou arrêtez les serveurs de commandes, les initiateurs de canal, les moniteurs de déclenchement et les programmes d'écoute.
- Définissez des services spécifiques pour démarrer automatiquement lorsqu'un gestionnaire de files d'attente est démarré.
- Modifiez les propriétés des gestionnaires de files d'attente.
- Modifiez le gestionnaire de files d'attente par défaut local.
- Créez des objets JMS à partir d'objets IBM MQ et des objets IBM MQ à partir d'objets JMS.
- Créez une JMS fabrique de connexions pour l'un des types actuellement pris en charge.
- Modifiez les paramètres d'un service, comme le numéro de port TCP d'un programme d'écoute ou le nom d'une file d'attente d'initiateur de canal.
- Démarrez ou arrêtez la trace de service.

### Vues de contenu et boîtes de dialogue de propriétés

Vous effectuez des tâches d'administration à l'aide d'une série de vues de contenu et de boîtes de dialogue de propriété.

#### Vue Contenu

Une vue de contenu est un panneau qui peut afficher les éléments suivants:

- Attributs et options d'administration relatifs à IBM MQ .
- Attributs et options d'administration relatifs à un ou plusieurs objets associés.
- Attributs et options d'administration d'un cluster.

## Boîtes de dialogue de propriétés

Une boîte de dialogue de propriété est un panneau qui affiche les attributs relatifs à un objet dans une série de zones, dont certaines peuvent être éditées.

Vous naviguez dans IBM MQ Explorer à l'aide de la vue Navigator . Le Navigator vous permet de sélectionner la vue de contenu dont vous avez besoin.

## Extension du IBM MQ Explorer

Le IBM MQ Explorer présente les informations dans un style cohérent avec celui de l'infrastructure Eclipse et les autres applications de plug-in prises en charge par Eclipse .

Grâce à l'extension du IBM MQ Explorer, les administrateurs système ont la possibilité de personnaliser le IBM MQ Explorer afin d'améliorer la manière dont ils administrent IBM MQ.

Pour plus d'informations, voir [Extension de MQ Explorer](#).

## Choix de l'utilisation du IBM MQ Explorer

Lorsque vous décidez d'utiliser le IBM MQ Explorer lors de votre installation, prenez en compte les informations répertoriées dans cette rubrique.

Vous devez prendre en compte les points suivants:

### Noms d'objet

Si vous utilisez des noms en minuscules pour les gestionnaires de files d'attente et les autres objets avec IBM MQ Explorer, lorsque vous utilisez les objets à l'aide des commandes MQSC, vous devez les placer entre apostrophes ou IBM MQ ne les reconnaît pas.

### Grands gestionnaires de files d'attente

IBM MQ Explorer fonctionne mieux avec les petits gestionnaires de files d'attente. Si vous disposez d'un grand nombre d'objets sur un seul gestionnaire de files d'attente, vous risquez de rencontrer des retards lors de l'extraction par IBM MQ Explorer des informations requises à présenter dans une vue.

### Clusters

Les clusters IBM MQ peuvent potentiellement contenir des centaines ou des milliers de gestionnaires de files d'attente. La IBM MQ Explorer présente les gestionnaires de files d'attente dans un cluster à l'aide d'une structure arborescente. La taille physique d'un cluster n'affecte pas de manière significative la vitesse du IBM MQ Explorer car IBM MQ Explorer ne se connecte pas aux gestionnaires de files d'attente du cluster tant que vous ne les avez pas sélectionnés.

## Configuration de IBM MQ Explorer

Cette section décrit les étapes à suivre pour configurer IBM MQ Explorer.

- [«Logiciels prérequis et définitions pour IBM MQ Explorer»](#), à la page 127
- [«Sécurité de IBM MQ Explorer»](#), à la page 128
- [«Affichage et masquage des gestionnaires de files d'attente et des clusters dans IBM MQ Explorer»](#), à la page 132
- [«Appartenance à un cluster et IBM MQ Explorer»](#), à la page 133
- [«Conversion de données pour IBM MQ Explorer»](#), à la page 133

## Logiciels prérequis et définitions pour IBM MQ Explorer

Vérifiez que vous respectez les exigences suivantes avant de tenter d'utiliser le IBM MQ Explorer.

IBM MQ Explorer peut se connecter à des gestionnaires de files d'attente éloignées à l'aide du protocole de communication TCP/IP uniquement.

Vérifiez que:

1. Un serveur de commandes est en cours d'exécution sur chaque gestionnaire de files d'attente administré à distance.

2. Un objet programme d'écoute TCP/IP approprié doit être en cours d'exécution sur chaque gestionnaire de files d'attente éloignées. Cet objet peut être le programme d'écoute IBM MQ ou, sur les systèmes AIX and Linux , le démon inetd.
3. Un canal de connexion serveur, par défaut nommé SYSTEM.ADMIN.SVRCONN, existe sur tous les gestionnaires de files d'attente éloignées.

Vous pouvez créer le canal à l'aide de la commande MQSC suivante:

```
DEFINE CHANNEL (SYSTEM.ADMIN.SVRCONN) CHLTYPE (SVRCONN)
```

Cette commande crée une définition de canal de base. Si vous souhaitez une définition plus sophistiquée (pour configurer la sécurité, par exemple), vous avez besoin de paramètres supplémentaires. Pour plus d'informations, voir [DEFINE CHANNEL](#).

4. La file d'attente système, SYSTEM.MQEXPLORER.REPLY.MODEL, doit exister.

## Sécurité de IBM MQ Explorer

Si vous utilisez IBM MQ dans un environnement où il est important que vous contrôliez l'accès des utilisateurs à des objets particuliers, vous devrez peut-être prendre en compte les aspects de sécurité de l'utilisation de IBM MQ Explorer.

### **Autorisation d'utiliser le IBM MQ Explorer**

Tout utilisateur peut utiliser le IBM MQ Explorer, mais certains droits sont requis pour la connexion, l'accès et la gestion des gestionnaires de files d'attente.

Pour effectuer des tâches d'administration locales à l'aide de IBM MQ Explorer, un utilisateur doit disposer des droits nécessaires pour effectuer les tâches d'administration. Si l'utilisateur est membre du groupe mqm , il est autorisé à effectuer toutes les tâches d'administration locales.

Pour se connecter à un gestionnaire de files d'attente éloignées et effectuer des tâches d'administration à distance à l'aide de IBM MQ Explorer, l'utilisateur exécutant IBM MQ Explorer doit disposer des droits suivants:

- Droits CONNECT sur l'objet gestionnaire de files d'attente cible
- Droits INQUIRE sur l'objet de gestionnaire de files d'attente cible
- Droits DISPLAY sur l'objet gestionnaire de files d'attente cible
- Droits INQUIRE sur la file d'attente, SYSTEM.MQEXPLORER.REPLY.MODEL
- Droits DISPLAY sur la file d'attente, SYSTEM.MQEXPLORER.REPLY.MODEL
- Droit d'entrée (get) sur la file d'attente, SYSTEM.MQEXPLORER.REPLY.MODEL
- Droit de sortie (put) sur la file d'attente, SYSTEM.MQEXPLORER.REPLY.MODEL
- Droit de sortie (put) sur la file d'attente, SYSTEM.ADMIN.COMMAND.QUEUE
- Droits INQUIRE sur la file d'attente, SYSTEM.ADMIN.COMMAND.QUEUE
- Droit d'exécution de l'action sélectionnée

**Remarque :** Les droits d'entrée sont liés à l'entrée à l'utilisateur à partir d'une file d'attente (opération d'extraction). Le droit OUTPUT est lié à la sortie de l'utilisateur vers une file d'attente (opération d'insertion).

Pour vous connecter à un gestionnaire de files d'attente éloignées sous IBM MQ for z/OS et effectuer des tâches d'administration à distance à l'aide de IBM MQ Explorer, les informations suivantes doivent être fournies:

- Un profil RACF pour la file d'attente système, SYSTEM.MQEXPLORER.REPLY.MODEL
- Un profil RACF pour les files d'attente, AMQ.MQEXPLORER. \*

En outre, l'utilisateur exécutant le IBM MQ Explorer doit disposer des droits suivants:

- RACF droits UPDATE sur la file d'attente système, SYSTEM.MQEXPLORER.REPLY.MODEL



- RACF droits UPDATE sur les files d'attente, AMQ.MQEXPLORER. \*
- Droits CONNECT sur l'objet gestionnaire de files d'attente cible
- Droit d'exécution de l'action sélectionnée
- Droits d'accès en lecture (READ) à tous les profils hlq.DISPLAY.object dans la classe MQCMD5

Pour plus d'informations sur l'octroi de droits sur des objets IBM MQ , voir [Attribution de droits d'accès à un objet IBM MQ sur des systèmes AIX, Linux, and Windows](#).

Si un utilisateur tente d'effectuer une opération qu'il n'est pas autorisé à effectuer, le gestionnaire de files d'attente cible appelle des procédures d'échec d'autorisation et l'opération échoue.

Le filtre par défaut dans IBM MQ Explorer consiste à afficher tous les objets IBM MQ . S'il existe des objets IBM MQ sur lesquels un utilisateur ne dispose pas des droits DISPLAY, des échecs d'autorisation sont générés. Si des événements de droits sont enregistrés, limitez la plage d'objets affichés aux objets sur lesquels l'utilisateur dispose des droits DISPLAY.

### ***Sécurité pour la connexion à des gestionnaires de files d'attente éloignées à partir de IBM MQ Explorer***

Vous devez sécuriser le canal entre IBM MQ Explorer et chaque gestionnaire de files d'attente éloignées.

IBM MQ Explorer se connecte aux gestionnaires de files d'attente éloignées en tant qu'application client MQI. Cela signifie que chaque gestionnaire de files d'attente éloignées doit disposer d'une définition de canal de connexion serveur et d'un programme d'écoute TCP/IP approprié. Si vous ne sécurisez pas votre canal de connexion serveur, une application malveillante peut se connecter au même canal de connexion serveur et accéder aux objets du gestionnaire de files d'attente avec des droits illimités. Pour sécuriser votre canal de connexion serveur, indiquez une valeur non vide pour l'attribut MCAUSER du canal, utilisez des enregistrements d'authentification de canal ou utilisez un exit de sécurité.

**La valeur par défaut de l'attribut MCAUSER est l'ID utilisateur local.** Si vous indiquez un nom d'utilisateur non vide comme attribut MCAUSER du canal de connexion serveur, tous les programmes qui se connectent au gestionnaire de files d'attente à l'aide de ce canal s'exécutent avec l'identité de l'utilisateur nommé et disposent du même niveau de droits d'accès. Cela ne se produit pas si vous utilisez des enregistrements d'authentification de canal.

### ***Utilisation d'un exit de sécurité avec IBM MQ Explorer***

Vous pouvez spécifier un exit de sécurité par défaut et des exits de sécurité spécifiques au gestionnaire de files d'attente à l'aide de IBM MQ Explorer.

Vous pouvez définir un exit de sécurité par défaut, qui peut être utilisé pour toutes les nouvelles connexions client à partir de IBM MQ Explorer. Cet exit par défaut peut être remplacé lors de l'établissement d'une connexion. Vous pouvez également définir un exit de sécurité pour un seul gestionnaire de files d'attente ou un ensemble de gestionnaires de files d'attente, qui prend effet lorsqu'une connexion est établie. Vous spécifiez des exits à l'aide de IBM MQ Explorer. Pour plus d'informations, reportez-vous à l'aide d'IBM MQ Explorer.

### ***Utilisation de IBM MQ Explorer pour la connexion à un gestionnaire de files d'attente éloignées à l'aide de canaux MQI TLS***

IBM MQ Explorer se connecte aux gestionnaires de files d'attente éloignées à l'aide d'un canal MQI. Si vous souhaitez sécuriser le canal MQI à l'aide de la sécurité TLS, vous devez établir le canal à l'aide d'une table de définition de canal du client.




Pour plus d'informations sur l'établissement d'un canal MQI à l'aide d'une table de définition de canal du client, voir [IBM MQ MQI clients](#).

Une fois que vous avez établi le canal à l'aide d'une table de définition de canal du client, vous pouvez utiliser IBM MQ Explorer pour vous connecter à un gestionnaire de files d'attente éloignées à l'aide d'un canal MQI activé par TLS, comme décrit dans «[Tâches sur le système qui héberge le gestionnaire de files d'attente éloignées](#)», à la page 130 et «[Tâches sur le système qui héberge le IBM MQ Explorer](#)», à la page 130.

## Tâches sur le système qui héberge le gestionnaire de files d'attente éloignées

Sur le système hébergeant le gestionnaire de files d'attente éloignées, effectuez les tâches suivantes:

1. Définissez une paire de canaux de connexion serveur et de connexion client et spécifiez la valeur appropriée pour l'attribut *SSLCIPH* sur la connexion serveur sur les deux canaux. Pour plus d'informations sur l'attribut *SSLCIPH*, voir [Protection des canaux avec TLS](#).
2. Envoyez la table de définition de canal AMQCLCHL .TAB, qui se trouve dans le répertoire @ipcc du gestionnaire de files d'attente, au système hébergeant IBM MQ Explorer.
3. Démarrez un programme d'écoute TCP/IP sur un port désigné.
4. Placez les certificats de l'autorité de certification et les certificats TLS personnels dans le répertoire SSL du gestionnaire de files d'attente:

-   /var/mqm/qmgrs/+QMNAME+/SSL pour les systèmes AIX and Linux .
-  C:\Program Files\IBM\MQ\qmgrs\+QMNAME+\SSL pour les systèmes Windows .

Où +QMNAME+ est un jeton représentant le nom du gestionnaire de files d'attente.

5. Créez un fichier de base de données de clés de type CMS nommé key .kdb. Stockez le mot de passe de la base de données de clés dans un fichier en spécifiant le paramètre -stash dans la commande **runmqakm** utilisée pour créer la base de données de clés.
6. Ajoutez les certificats de l'autorité de certification à la base de données de clés créée à l'étape précédente.
7. Importez le certificat personnel du gestionnaire de files d'attente dans la base de données de clés.

Pour plus d'informations sur l'utilisation de TLS sur les systèmes Windows, voir [Utilisation de TLS sur AIX, Linux, and Windows](#).

## Tâches sur le système qui héberge le IBM MQ Explorer

Sur le système hébergeant le IBM MQ Explorer, effectuez les tâches suivantes:

1. Créez un fichier de base de données de clés de type JKS nommé key .jks. Définissez un mot de passe pour ce fichier de base de données de clés.  
  
Le magasin de clés utilisé par IBM MQ Explorer pour la sécurité TLS doit être un fichier de magasin de clés Java (JKS).
2. Ajoutez les certificats de l'autorité de certification à la base de données de clés créée à l'étape précédente.
3. Importez le certificat personnel du gestionnaire de files d'attente dans la base de données de clés.
4. Sur les systèmes Windows et Linux, démarrez IBM MQ Explorer à l'aide du menu système, du fichier exécutable MQExplorer ou de la commande **strmqcfcg**.
5. Dans la barre d'outils IBM MQ Explorer, cliquez sur **Fenêtre-> Préférences**, puis développez **IBM MQ Explorer** et cliquez sur **Magasins de certificats client SSL**. Entrez le nom et le mot de passe du fichier JKS créé à l'étape 1 de «Tâches sur le système qui héberge le IBM MQ Explorer», à la page 130, dans le magasin de certificats de confiance et dans le magasin de certificats personnels, puis cliquez sur **OK**.
6. Fermez la fenêtre **Préférences** et cliquez avec le bouton droit de la souris sur **Gestionnaires de files d'attente**. Cliquez sur **Afficher / Masquer les gestionnaires de files d'attente**, puis sur **Ajouter** dans l'écran **Afficher / Masquer les gestionnaires de files d'attente**.
7. Entrez le nom du gestionnaire de files d'attente et sélectionnez l'option **Se connecter directement**. Cliquez sur Next.
8. Sélectionnez **Utiliser la table de définition de canal du client (CCDT)** et indiquez l'emplacement du fichier de table de canaux que vous avez transféré du gestionnaire de files d'attente éloignées à l'étape 2 dans «Tâches sur le système qui héberge le gestionnaire de files d'attente éloignées», à la page 130 sur le système hébergeant le gestionnaire de files d'attente éloignées.

9. Cliquez sur **Terminer**. Vous pouvez maintenant accéder au gestionnaire de files d'attente éloignées à partir de la IBM MQ Explorer.

### **Connexion via un autre gestionnaire de files d'attente avec IBM MQ Explorer**

IBM MQ Explorer vous permet de vous connecter à un gestionnaire de files d'attente via un gestionnaire de files d'attente intermédiaire, auquel IBM MQ Explorer est déjà connecté.

Dans ce cas, IBM MQ Explorer insère des messages de commande PCF dans le gestionnaire de files d'attente intermédiaire, en spécifiant ce qui suit:

- Le paramètre *ObjectQMgrName* dans le descripteur d'objet (MQOD) comme nom du gestionnaire de files d'attente cible. Pour plus d'informations sur la résolution de nom de file d'attente, voir [Résolution de nom](#).
- Le paramètre *UserIdentifier* dans le descripteur de message (MQMD) en tant qu' *userIdlocal*.

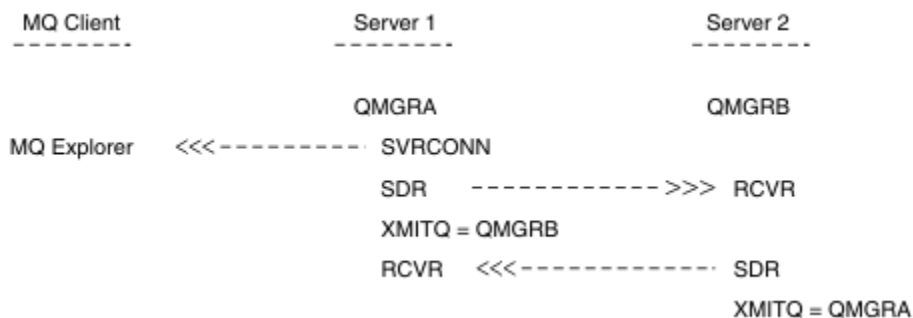
Si la connexion est ensuite utilisée pour se connecter au gestionnaire de files d'attente cible via un gestionnaire de files d'attente intermédiaire, *userId* est de nouveau transmis dans le paramètre *UserIdentifier* du descripteur de message (MQMD). Pour que le programme d'écoute MCA sur le gestionnaire de files d'attente cible accepte ce message, l'attribut MCAUSER doit être défini ou l' *userId* doit déjà exister avec le droit d'insertion.

Le serveur de commandes du gestionnaire de files d'attente cible insère des messages dans la file d'attente de transmission en spécifiant *userId* dans le paramètre *UserIdentifier* du descripteur de message (MQMD). Pour que cette insertion aboutisse, l' *userId* doit déjà exister sur le gestionnaire de files d'attente cible avec les droits d'insertion.

L'exemple suivant montre comment connecter un gestionnaire de files d'attente, via un gestionnaire de files d'attente intermédiaire, à IBM MQ Explorer.

Etablissez une connexion d'administration à distance à un gestionnaire de files d'attente. Vérifiez que:

- Le gestionnaire de files d'attente sur le serveur est actif et un canal de connexion serveur (SVRCONN) est défini.
- Le programme d'écoute est actif.
- Le serveur de commandes est actif.
- SYSTEME.SYSTEM.MQ\_EXPLORER.REPLY.MODEL a été créée et vous disposez des droits suffisants.
- Les programmes d'écoute du gestionnaire de files d'attente, les serveurs de commandes et les canaux émetteurs sont démarrés.



Dans cet exemple :

- IBM MQ Explorer est connecté au gestionnaire de files d'attente QMGRA (exécuté sur Server1) à l'aide d'une connexion client.
- Le gestionnaire de files d'attente QMGRB sur Server2 peut désormais être connecté à IBM MQ Explorer via un gestionnaire de files d'attente intermédiaire ( QMGRA)
- Lors de la connexion à QMGRB avec IBM MQ Explorer, sélectionnez QMGRA comme gestionnaire de files d'attente intermédiaire

Dans cette situation, il n'y a pas de connexion directe à QMGRB à partir de IBM MQ Explorer; la connexion à QMGRB s'effectue via QMGRA.

Le gestionnaire de files d'attente QMGRB sur Server2 est connecté à QMGRA sur Server1 à l'aide de canaux émetteurs-récepteurs. Le canal entre QMGRA et QMGRB doit être configuré de sorte que l'administration à distance soit possible ; voir «Configuration des gestionnaires de files d'attente pour l'administration à distance», à la page 207.

## Affichage et masquage des gestionnaires de files d'attente et des clusters dans IBM MQ Explorer

IBM MQ Explorer peut afficher plusieurs gestionnaires de files d'attente à la fois. Dans le panneau Afficher / Masquer le gestionnaire de files d'attente (sélectionnable dans le menu du noeud d'arborescence des gestionnaires de files d'attente), vous pouvez choisir d'afficher ou non des informations sur une autre machine (distante). Les gestionnaires de files d'attente locaux sont détectés automatiquement.

Pour afficher un gestionnaire de files d'attente éloignées:

1. Cliquez avec le bouton droit de la souris sur le noeud d'arborescence **Gestionnaires de files d'attente** , puis sélectionnez **Afficher / Masquer les gestionnaires de files d'attente**.
2. Cliquez sur **Ajouter**. Le panneau Afficher / Masquer les gestionnaires de files d'attente s'affiche.
3. Entrez le nom du gestionnaire de files d'attente éloignées et le nom d'hôte ou l'adresse IP dans les zones fournies.

Le nom d'hôte ou l'adresse IP est utilisé pour établir une connexion client au gestionnaire de files d'attente éloignées à l'aide de son canal de connexion serveur par défaut, SYSTEM.ADMIN.SVRCONN, ou un canal de connexion serveur défini par l'utilisateur.

4. Cliquez sur **Terminer**.

Le panneau Afficher / Masquer les gestionnaires de files d'attente affiche également la liste de tous les gestionnaires de files d'attente visibles. Vous pouvez utiliser ce panneau pour masquer les gestionnaires de files d'attente dans la vue de navigation.

Si IBM MQ Explorer affiche un gestionnaire de files d'attente qui est membre d'un cluster, le cluster est détecté et affiché automatiquement.

Pour exporter la liste des gestionnaires de files d'attente éloignées à partir de ce panneau:

1. Fermez le panneau Afficher / Masquer les gestionnaires de files d'attente.
2. Cliquez avec le bouton droit de la souris sur le noeud d'arborescence **IBM MQ** le plus élevé dans la sous-fenêtre de navigation du IBM MQ Explorer, puis sélectionnez **Exporter les paramètres IBM MQ Explorer** .
3. Cliquez sur **IBM MQ Explorer > IBM MQ Explorer Paramètres**
4. Sélectionnez **Informations de connexion > Gestionnaires de files d'attente éloignées**.
5. Sélectionnez un fichier dans lequel stocker les paramètres exportés.
6. Enfin, cliquez sur **Terminer** pour exporter les informations de connexion du gestionnaire de files d'attente éloignées dans le fichier spécifié.

Pour importer une liste de gestionnaires de files d'attente éloignées:

1. Cliquez avec le bouton droit de la souris sur le noeud d'arborescence **IBM MQ** le plus élevé dans la sous-fenêtre de navigation du IBM MQ Explorer, puis sélectionnez **Importer les paramètres IBM MQ Explorer** .
2. Cliquez sur **IBM MQ Explorer > IBM MQ Explorer Paramètres**
3. Cliquez sur **Parcourir** et accédez au chemin du fichier contenant les informations de connexion du gestionnaire de files d'attente éloignées.
4. Cliquez sur **Ouvrir**. Si le fichier contient une liste de gestionnaires de files d'attente éloignées, la case **Informations de connexion > Gestionnaires de files d'attente éloignées** est cochée.

- Enfin, cliquez sur **Terminer** pour importer les informations de connexion du gestionnaire de files d'attente éloignées dans le IBM MQ Explorer.

## Appartenance à un cluster et IBM MQ Explorer

IBM MQ Explorer requiert des informations sur les gestionnaires de files d'attente qui sont membres d'un cluster.

Si un gestionnaire de files d'attente est membre d'un cluster, le noeud de l'arborescence de clusters est renseigné automatiquement.

Si les gestionnaires de files d'attente deviennent membres de clusters alors que le IBM MQ Explorer est en cours d'exécution, vous devez gérer le IBM MQ Explorer avec des données d'administration à jour sur les clusters afin qu'il puisse communiquer efficacement avec eux et afficher des informations de cluster correctes lorsque cela est demandé. Pour ce faire, IBM MQ Explorer a besoin des informations suivantes:

- Nom d'un gestionnaire de files d'attente de référentiel
- Nom de connexion du gestionnaire de files d'attente de référentiel s'il se trouve sur un gestionnaire de files d'attente éloignées

Avec ces informations, IBM MQ Explorer peut:

- Utilisez le gestionnaire de files d'attente de référentiel pour obtenir la liste des gestionnaires de files d'attente du cluster.
- Administrez les gestionnaires de files d'attente qui sont membres du cluster et qui se trouvent sur des plateformes et des niveaux de commande pris en charge.

L'administration n'est pas possible si:

- Le référentiel choisi devient indisponible. IBM MQ Explorer ne bascule pas automatiquement vers un autre référentiel.
- Le référentiel choisi ne peut pas être contacté via TCP/IP.
- Le référentiel choisi s'exécute sur un gestionnaire de files d'attente qui s'exécute sur une plateforme et dont le niveau de commande n'est pas pris en charge par IBM MQ Explorer.

Les membres du cluster qui peuvent être administrés peuvent être locaux ou distants s'ils peuvent être contactés à l'aide de TCP/IP. IBM MQ Explorer se connecte aux gestionnaires de files d'attente locaux qui sont membres d'un cluster directement, sans utiliser de connexion client.

## Conversion de données pour IBM MQ Explorer

IBM MQ Explorer fonctionne avec le CCSID 1208 (UTF-8). Cela permet à IBM MQ Explorer d'afficher correctement les données des gestionnaires de files d'attente éloignées. Que vous vous connectiez à un gestionnaire de files d'attente directement ou à l'aide d'un gestionnaire de files d'attente intermédiaire, IBM MQ Explorer requiert la conversion de tous les messages entrants au CCSID 1208 (UTF-8).

Un message d'erreur est émis si vous tentez d'établir une connexion entre IBM MQ Explorer et un gestionnaire de files d'attente dont le CCSID n'est pas reconnu par IBM MQ Explorer .

Les conversions prises en charge sont décrites dans la rubrique [Conversion de page de codes](#).

## Utilisation de l'application IBM MQ Taskbar (Windows uniquement)

L'application IBM MQ Taskbar affiche une icône dans la barre des tâches système Windows sur le serveur. L'icône vous fournit le statut en cours de IBM MQ et un menu à partir duquel vous pouvez effectuer des actions simples.

Sous Windows, l'icône IBM MQ se trouve dans la barre des tâches système du serveur et est remplacée par un symbole de statut codé par des couleurs, qui peut avoir l'une des significations suivantes:

### **Vert**

Fonctionnement correct ; aucune alerte à l'heure actuelle

### **Bleu**

Indéterminé ; IBM MQ est en cours de démarrage ou d'arrêt

### **Jaune**

Alerte ; un ou plusieurs services échouent ou ont déjà échoué

Pour afficher le menu, cliquez avec le bouton droit de la souris sur l'icône IBM MQ . Dans le menu, vous pouvez effectuer les actions suivantes:

- Cliquez sur **Ouvrir** pour ouvrir le moniteur d'alertes IBM MQ .
- Cliquez sur **Quitter** pour quitter l'application IBM MQ Taskbar.
- Cliquez sur **IBM MQ Explorer** pour démarrer le IBM MQ Explorer.
- Cliquez sur **Arrêter IBM MQ** pour arrêter IBM MQ.
- Cliquez sur **A propos de IBM MQ** pour afficher des informations sur le moniteur d'alertes IBM MQ .

## **L'application du moniteur d'alertes IBM MQ ( Windows uniquement)**

Le moniteur d'alertes IBM MQ est un outil de détection des erreurs qui identifie et enregistre les problèmes liés à IBM MQ sur une machine locale.

Le moniteur d'alertes affiche des informations sur le statut en cours de l'installation locale d'un serveur IBM MQ . Il surveille également l'interface ACPI ( Windows Advanced Configuration and Power Interface) et s'assure que les paramètres ACPI sont appliqués.

A partir du moniteur d'alertes IBM MQ , vous pouvez:

- Accès direct à IBM MQ Explorer
- Afficher les informations relatives à toutes les alertes en attente
- Arrêt du service IBM MQ sur la machine locale
- Acheminer les messages d'alerte sur le réseau vers un compte utilisateur configurable ou vers un poste de travail ou un serveur Windows

## **Utilisation des objets IBM MQ locaux**

---

Vous pouvez administrer des objets IBM MQ locaux pour prendre en charge les programmes d'application qui utilisent l'interface MQI (Message Queue Interface).

### **Pourquoi et quand exécuter cette tâche**

Dans ce contexte, l'administration locale implique la création, l'affichage, la modification, la copie et la suppression d'objets IBM MQ .

Outre les approches décrites dans cette section, vous pouvez utiliser IBM MQ Explorer pour administrer des objets IBM MQ locaux. Pour plus d'informations, voir [«Administration à l'aide de IBM MQ Explorer»](#), à la page 125.

### **Procédure**

- Utilisez les informations des rubriques suivantes pour vous aider à administrer les objets IBM MQ locaux.
  - [Programmes d'application utilisant l'interface MQI](#)
  - [«Administration d' IBM MQ à l'aide de commandes MQSC»](#), à la page 12
  - [«Affichage et modification des attributs du gestionnaire de files d'attente»](#), à la page 143
  - [«Utilisation des files d'attente locales»](#), à la page 146

- «Utilisation de files d'attente alias», à la page 160
- «Utilisation des files d'attente modèles», à la page 162
- «Utilisation des services», à la page 192
- «Gestion des objets pour le déclenchement», à la page 200

## Utilisation des gestionnaires de files d'attente

Vous pouvez utiliser des commandes de contrôle pour démarrer et arrêter un gestionnaire de files d'attente. Vous pouvez utiliser des commandes MQSC pour afficher ou modifier des attributs de gestionnaire de files d'attente.

### Tâches associées

[Création de gestionnaires de files d'attente sur Multiplatforms](#)

### Multi Démarrage d'un gestionnaire de files d'attente

Lorsque vous créez un gestionnaire de files d'attente, vous devez le démarrer pour lui permettre de traiter des commandes ou des appels MQI.

### Pourquoi et quand exécuter cette tâche

Vous pouvez démarrer un gestionnaire de files d'attente à l'aide de la commande **strmqm**. Pour une description de la commande **strmqm** et de ses options, voir [strmqm](#).

**Windows** **Linux** Sur les systèmes Windows et Linux (plateformes x86 et x86-64), vous pouvez également démarrer un gestionnaire de files d'attente à l'aide de la IBM MQ Explorer.

**Windows** Sous Windows, vous pouvez démarrer automatiquement un gestionnaire de files d'attente lorsque le système démarre à l'aide de la IBM MQ Explorer. Pour plus d'informations, voir «Administration à l'aide de IBM MQ Explorer», à la page 125.

### Procédure

- Pour démarrer un gestionnaire de files d'attente à l'aide de la commande **strmqm**, entrez la commande suivie du nom du gestionnaire de files d'attente à démarrer.  
Par exemple, pour démarrer un gestionnaire de files d'attente appelé QMB, entrez la commande suivante:

```
strmqm QMB
```

**Remarque :** Vous devez utiliser la commande **strmqm** à partir de l'installation associée au gestionnaire de files d'attente utilisé. Vous pouvez déterminer à quelle installation un gestionnaire de files d'attente est associé à l'aide de la commande `dspmq -o installation`.

La commande **strmqm** ne renvoie pas de contrôle tant que le gestionnaire de files d'attente n'a pas démarré et n'est pas prêt à accepter les demandes de connexion.

- **Windows** **Linux**  
Pour démarrer un gestionnaire de files d'attente à l'aide de la console IBM MQ Explorer, procédez comme suit:
  - a) Ouvrez IBM MQ Explorer.
  - b) Dans la vue Navigator, sélectionnez le gestionnaire de files d'attente.
  - c) Cliquez sur **Démarrer**.

### Résultats

Le gestionnaire de files d'attente démarre.

Si le démarrage du gestionnaire de files d'attente prend plus de quelques secondes, IBM MQ émet des messages d'information détaillant par intermittence la progression du démarrage.

Multi

## Arrêt d'un gestionnaire de files d'attente

Vous pouvez utiliser la commande **endmqm** pour arrêter un gestionnaire de files d'attente. Cette commande permet d'arrêter un gestionnaire de files d'attente de quatre manières: un arrêt contrôlé ou mis au repos, un arrêt immédiat, un arrêt préventif et un arrêt en attente. Vous pouvez également, sous Windows et Linux, arrêter un gestionnaire de files d'attente à l'aide de la IBM MQ Explorer.

### Pourquoi et quand exécuter cette tâche

Il existe quatre façons d'arrêter un gestionnaire de files d'attente à instance unique à l'aide de la commande **endmqm** :

#### Arrêt contrôlé (mis au repos)

Par défaut, la commande **endmqm** effectue un arrêt au repos du gestionnaire de files d'attente spécifié. Un arrêt mis au repos attend que toutes les applications connectées se soient déconnectées, ce qui peut prendre un certain temps.

#### Arrêt immédiat

Pour un arrêt immédiat, tous les appels MQI en cours sont autorisés à se terminer, mais tous les nouveaux appels échouent. Ce type d'arrêt n'attend pas que les applications se déconnectent du gestionnaire de files d'attente.

#### Arrêt préemptif

Le gestionnaire de files d'attente s'arrête immédiatement. Utilisez ce type d'arrêt uniquement dans des circonstances exceptionnelles, par exemple, lorsqu'un gestionnaire de files d'attente ne s'arrête pas suite à une commande **endmqm** normale.

#### Attente de l'arrêt

Ce type d'arrêt est équivalent à un arrêt contrôlé, sauf que le contrôle vous est renvoyé uniquement après l'arrêt du gestionnaire de files d'attente.

La commande **endmqm** arrête toutes les instances d'un gestionnaire de files d'attente multi-instance de la même manière qu'elle arrête un gestionnaire de files d'attente à instance unique. Vous pouvez émettre la commande **endmqm** sur l'instance active ou sur l'une des instances de secours d'un gestionnaire de files d'attente multi-instance. Toutefois, vous devez exécuter **endmqm** sur l'instance active pour arrêter le gestionnaire de files d'attente.

Vous avez la possibilité d'arrêter le gestionnaire de files d'attente dans un délai cible d'un nombre de secondes que vous spécifiez, avec ou sans interruption des tâches de maintenance du gestionnaire de files d'attente non essentielles. Voir «Arrêt d'un gestionnaire de files d'attente dans un délai cible», à la page 138.



#### Avertissement :

- Les messages persistants sont conservés quel que soit le type d'arrêt utilisé (y compris l'arrêt manuel des processus IBM MQ ), alors que les messages non persistants ne peuvent pas survivre à tout type d'arrêt.

La spécification de la propriété de file d'attente NPMCLASS (HIGH) permet de sauvegarder les messages non persistants de manière optimale. L'utilisation de processus **endmqm -t**, **endmqm -tp**, **endmqm -p** ou l'arrêt manuel de IBM MQ réduit les risques que les messages NPMCLASS (HIGH) survivent à un cycle d'arrêt ou de redémarrage IBM MQ par rapport à **endmqm -w** ou **endmqm -i**

- La durée combinée de l'arrêt et du redémarrage du gestionnaire de files d'attente peut être plus longue en raison de l'utilisation d'une méthode d'arrêt plus brusque, en particulier lors de l'utilisation des options **-p** et **-tp**.

Si le gestionnaire de files d'attente doit recourir à l'arrêt des processus IBM MQ pour arrêter le gestionnaire de files d'attente, une synchronisation plus poussée de l'état du gestionnaire





de files d'attente est probablement nécessaire lorsque le gestionnaire de files d'attente est redémarré.

Pour une description détaillée de la commande **endmqm** et de ses options, voir [endmqm](#).

**Conseil :** Les problèmes liés à l'arrêt d'un gestionnaire de files d'attente sont souvent dus à des applications. Par exemple, lorsque des applications:

- Ne pas vérifier correctement les codes retour MQI
- Ne pas demander de notification de mise au repos
- Arrêter sans se déconnecter du gestionnaire de files d'attente (en émettant un appel MQDISC )

Si un problème se produit lorsque vous tentez d'arrêter le gestionnaire de files d'attente, vous pouvez sortir de la commande **endmqm** en utilisant Ctrl-C. Vous pouvez ensuite exécuter une autre commande **endmqm** , mais cette fois avec un paramètre qui spécifie le type d'arrêt dont vous avez besoin.

  Comme alternative à l'utilisation de la commande **endmqm** , sous Windows et Linux, vous pouvez arrêter un gestionnaire de files d'attente à l'aide de la IBM MQ Explorer pour effectuer un arrêt contrôlé ou immédiat.

## Procédure

- Pour arrêter le gestionnaire de files d'attente à l'aide de la commande **endmqm** , entrez la commande suivie du paramètre approprié, si nécessaire, et du nom du gestionnaire de files d'attente à arrêter.

**Remarque :** Vous devez utiliser la commande **endmqm** à partir de l'installation associée au gestionnaire de files d'attente que vous utilisez. Pour savoir à quelle installation un gestionnaire de files d'attente est associé, utilisez la commande [dspmq](#) :

```
dspmq -o installation
```

- Pour effectuer un arrêt contrôlé (mis au repos), entrez la commande **endmqm** comme illustré dans l'exemple suivant, qui arrête un gestionnaire de files d'attente appelé QMB:

```
endmqm QMB
```

Sinon, la saisie de la commande **endmqm** avec le paramètre **-c** , comme illustré dans l'exemple suivant, est équivalente à une commande `endmqm QMB` .

```
endmqm -c QMB
```

Dans les deux cas, le contrôle vous est rendu immédiatement et vous n'êtes pas averti lorsque le gestionnaire de files d'attente s'est arrêté. Si vous souhaitez que la commande attende que toutes les applications se soient arrêtées et que le gestionnaire de files d'attente se soit arrêté avant de vous renvoyer le contrôle, utilisez le paramètre **-w** à la place, comme illustré dans l'exemple suivant.

```
endmqm -w QMB
```

- Pour effectuer un arrêt immédiat, entrez la commande **endmqm** avec le paramètre **-i** comme illustré dans l'exemple suivant:

```
endmqm -i QMB
```

- Pour effectuer un arrêt préventif, entrez la commande **endmqm** avec le paramètre **-p** comme illustré dans l'exemple suivant:

```
endmqm -p QMB
```



**Avertissement :** Un arrêt préventif peut avoir des conséquences imprévisibles pour les applications connectées. N'utilisez cette option que si toutes les autres tentatives d'arrêt du gestionnaire de files d'attente à l'aide d'une commande **endmqm** normale ont échoué.

**ALW**

Si l'arrêt préventif ne fonctionne pas, essayez «Arrêt manuel d'un gestionnaire de files d'attente», à la page 139 à la place.

- Pour demander la reconnexion automatique du client, entrez la commande **endmqm** avec le paramètre **-r**. Ce paramètre a pour effet de rétablir la connectivité des clients aux autres gestionnaires de files d'attente de leur groupe de gestionnaires de files d'attente.

**Remarque :** L'arrêt d'un gestionnaire de files d'attente à l'aide de la commande **endmqm** par défaut ne déclenche pas la reconnexion automatique du client.

- Pour effectuer un transfert vers une instance de secours d'un gestionnaire de files d'attente multi-instance après avoir arrêté l'instance active, entrez la commande **endmqm** avec le paramètre **-s** sur l'instance active du gestionnaire de files d'attente multi-instance.
- Pour arrêter l'instance de secours d'un gestionnaire de files d'attente multi-instance et laisser l'instance active en cours d'exécution, entrez la commande **endmqm** avec le paramètre **-x** sur l'instance de secours du gestionnaire de files d'attente multi-instance.

**Windows**

**Linux**

Sous Windows et Linux, pour arrêter le gestionnaire de files d'attente à l'aide d' IBM MQ Explorer, procédez comme suit:

- a) Ouvrez IBM MQ Explorer.
- b) Sélectionnez le gestionnaire de files d'attente dans la vue Navigator .
- c) Cliquez sur **Arrêter**.  
Le panneau **End Queue Manager** s'affiche.
- d) Sélectionnez **Contrôlé** ou **Immédiat**.
- e) Cliquez sur **OK**.  
Le gestionnaire de files d'attente s'arrête.

### Tâches associées

[Application de mises à jour du niveau de maintenance à des gestionnaires de files d'attente multi-instances sous AIX](#)

[Application de mises à jour du niveau de maintenance à des gestionnaires de files d'attente multi-instances sous Linux](#)

[Application de mises à jour du niveau de maintenance à des gestionnaires de files d'attente multi-instances sous Windows](#)

### Référence associée

[endmqm \(arrêt d'un gestionnaire de files d'attente\)](#)

### **Arrêt d'un gestionnaire de files d'attente dans un délai cible**

Vous pouvez arrêter le gestionnaire de files d'attente dans un délai cible de quelques secondes que vous spécifiez, avec ou sans interruption des tâches de maintenance de gestionnaire de files d'attente non essentielles.

Il existe deux façons de spécifier une heure cible lorsque vous utilisez la commande **endmqm**. L'option **-t** permet à toutes les tâches de maintenance du gestionnaire de files d'attente de se terminer, ce qui peut prolonger la phase d'arrêt du gestionnaire de files d'attente. L'option **-tp** interrompt les tâches de maintenance de gestionnaire de files d'attente non essentielles si nécessaire pour respecter l'heure cible spécifiée.

Les tâches de maintenance non essentielles incluent: la compression des fichiers de file d'attente et la persistance des messages NPMCLASS (HIGH). Dans le reste de cette page, le mot "ménage" est utilisé.

En fonction des modèles d'utilisation de votre application, la compression des fichiers de file d'attente peut prendre beaucoup de temps. Par conséquent, si votre objectif principal est d'arrêter rapidement le gestionnaire de files d'attente, utilisez l'option **-tp**.

Lorsque vous spécifiez une heure cible, le type d'arrêt **-w**, **-i** ou **-p** indique le type d'arrêt de démarrage.

**Remarque :** Un arrêt de `immediate` est toujours ordonné et diffère d'un arrêt de `controlled` principalement par la manière dont les applications en cours d'exécution sont mises au repos. Un arrêt de `immediate` effectue toujours des opérations de nettoyage. Un arrêt limité dans le temps annule ces actions lorsqu'elles interfèrent avec l'atteinte de l'heure cible.

Le gestionnaire de files d'attente transmet le type d'arrêt si nécessaire, afin d'atteindre l'heure cible. Exemple :

- Une cible **-t** de 10 secondes démarrant à **-w** peut être une mise au repos de sept secondes, un arrêt immédiat du gestionnaire de files d'attente de deux secondes, avec nettoyage, puis un arrêt immédiat sans nettoyage supplémentaire:

```
endmqm -w -t 10 queue_manager
```

- Une cible **-tp** de 10 secondes peut être une mise au repos de sept secondes, un arrêt immédiat du gestionnaire de files d'attente de deux secondes, avec nettoyage, un arrêt immédiat d'une seconde sans nettoyage supplémentaire, puis l'arrêt des processus IBM MQ :

```
endmqm -c -tp 10 queue_manager
```

- Une cible **-tp** de deux secondes à **-i** peut être un arrêt immédiat d'une seconde du gestionnaire de files d'attente, y compris le nettoyage, un arrêt immédiat d'une seconde sans nettoyage supplémentaire, puis commencer à arrêter les processus IBM MQ :

```
endmqm -i -tp 2 queue_manager
```

- Une cible d'une seconde à **-w** peut être 0.1 secondes à `wait`, par exemple, juste assez longtemps pour envoyer des codes retour IBM MQ aux applications connectées, 0.9 secondes d'arrêt immédiat du gestionnaire de files d'attente, y compris la maintenance, puis l'arrêt immédiat sans autre maintenance ; puis arrêter les processus IBM MQ .

### Référence associée

[endmqm \(arrêt d'un gestionnaire de files d'attente\)](#)


## Arrêt manuel d'un gestionnaire de files d'attente

Si les méthodes standard d'arrêt et de suppression d'un gestionnaire de files d'attente échouent, vous pouvez essayer d'arrêter le gestionnaire de files d'attente manuellement.

### Pourquoi et quand exécuter cette tâche

La méthode standard d'arrêt des gestionnaires de files d'attente consiste à utiliser la commande **endmqm**, comme décrit dans [«Arrêt d'un gestionnaire de files d'attente»](#), à la page 136. Si vous ne parvenez pas à arrêter un gestionnaire de files d'attente de manière standard, vous pouvez essayer d'arrêter un gestionnaire de files d'attente manuellement. La manière dont vous effectuez cette opération dépend de la plateforme que vous utilisez.

### Procédure

-  Pour arrêter un gestionnaire de files d'attente sous Windows, voir [«Arrêt manuel d'un gestionnaire de files d'attente sous Windows»](#), à la page 140.

-  

Pour arrêter un gestionnaire de files d'attente sous AIX ou Linux, voir [«Arrêt manuel d'un gestionnaire de files d'attente sous AIX and Linux»](#), à la page 141.

### Tâches associées

[Création et gestion des gestionnaires de files d'attente sur Multiplatforms](#)

### Référence associée

[endmqm](#)

### **Arrêt manuel d'un gestionnaire de files d'attente sous Windows**

Si vous ne parvenez pas à arrêter un gestionnaire de files d'attente sous Windows à l'aide de la commande **endmqm**, vous pouvez essayer d'arrêter le gestionnaire de files d'attente manuellement en arrêtant tous les processus en cours d'exécution et en arrêtant le service IBM MQ.

### Pourquoi et quand exécuter cette tâche

**Conseil :** Le gestionnaire de tâches Windows et la commande **tasklist** fournissent des informations limitées sur les tâches. Pour plus d'informations permettant de déterminer quels processus sont liés à un gestionnaire de files d'attente particulier, utilisez un outil tel que *Process Explorer* (`procxp.exe`), qui peut être téléchargé à partir du site Web Microsoft à l'adresse <http://www.microsoft.com>.

Pour arrêter un gestionnaire de files d'attente sous Windows, procédez comme suit.

### Procédure

1. Répertoriez les noms (ID) des processus en cours d'exécution à l'aide du gestionnaire de tâches Windows.
2. Arrêtez les processus à l'aide du gestionnaire de tâches Windows ou de la commande **taskkill**, dans l'ordre suivant (s'ils sont en cours d'exécution):

Nom du processus	Description
AMQZMUC0	Gestionnaire de processus critique
AMQZXMA0	Contrôleur d'exécution
AMQZFUMA	Processus OAM
AMQZLAA0	Agents LQM
AMQZLSA0	Agents LQM
AMQZMUFO	Gestionnaire d'utilitaire
AMQZMGRO	Ctrlur de processus
AMQZMUR0	Gestionnaire de processus redémarrable
AMQFQ PUB	Processus de publication / d'abonnement
AMQFCXBA	Processus de travail de courtier
AMQRMPPA	Processus de regroupement de processus
amqcrsta	Processus de travail de répondeur non à unités d'exécution
AMQCRS6B	Connexion client et canal récepteur LU62
AMQRRMFA	Processus de référentiel (pour les clusters)
AMQPCSEA	Serveur de commandes

Tableau 7. Processus Windows à arrêter en cas d'exécution (suite)	
Nom du processus	Description
runmqtrm	Appel d'un moniteur de déclenchement pour un serveur
runmqdlq	Appeler le gestionnaire de files d'attente de rebut
runmqchi	Processus de l'initiateur de canal
runmqslr	Processus du programme d'écoute de canal
AMQXSSVN	Serveurs de mémoire partagée

3. Arrêtez le service IBM MQ à partir des **outils d'administration** > **Services** du panneau de configuration Windows .
4. Si vous avez essayé toutes les méthodes et que le gestionnaire de files d'attente ne s'est pas arrêté, réamorcer votre système.

## Linux > AIX Arrêt manuel d'un gestionnaire de files d'attente sous AIX and Linux

Si vous ne parvenez pas à arrêter un gestionnaire de files d'attente sous AIX ou Linux à l'aide de la commande **endmqm** , vous pouvez essayer d'arrêter le gestionnaire de files d'attente manuellement en arrêtant tous les processus en cours d'exécution et en arrêtant le service IBM MQ .

### Pourquoi et quand exécuter cette tâche

Pour arrêter un gestionnaire de files d'attente sous AIX ou Linux, procédez comme suit.

Si vous arrêtez le gestionnaire de files d'attente manuellement, FFST peut être utilisé et les fichiers FDC placés dans `/var/mqm/errors`. Il ne doit pas être considéré comme un incident dans le gestionnaire de files d'attente.

Le gestionnaire de files d'attente redémarre normalement, même après que vous l'ayez arrêté à l'aide de cette méthode d'arrêt manuel.

### Procédure

1. Recherchez les identificateurs de processus (PID) des programmes de gestionnaire de files d'attente qui sont toujours en cours d'exécution à l'aide de la commande **ps** .

Par exemple, si le gestionnaire de files d'attente est appelé QMNAME, utilisez la commande suivante:

```
ps -ef | grep QMNAME
```

2. Arrêtez tous les processus de gestionnaire de files d'attente qui sont toujours en cours d'exécution à l'aide de la commande **kill** , en spécifiant les PID détectés à l'aide de la commande **ps** .

Pour mettre fin à un processus, utilisez **kill -KILL <pid>** ou la commande **kill -9 <pid>** équivalente.

Vous devez utiliser les PID que vous voulez tuer, un par un, en émettant cette commande à chaque fois.

**Important :** Si vous utilisez un signal autre que **9 (SIGKILL)** , le processus ne s'arrêtera probablement pas et vous obtiendrez des résultats imprévisibles.

Arrêtez les processus dans l'ordre suivant:

<i>Tableau 8. Processus AIX and Linux à arrêter en cas d'exécution</i>	
<b>Nom du processus</b>	<b>Description</b>
amqzmuc0	Gestionnaire de processus critique
amqzxma0	Contrôleur d'exécution
amqzfuma	Processus OAM
amqzlaa0	Agents LQM
amqzlsa0	Agents LQM
amqzmuf0	Gestionnaire d'utilitaire
amqzmur0	Gestionnaire de processus redémarrable
amqzmgr0	Ctrlur de processus
amqfqpub	Processus de publication / d'abonnement
amqfcxba	Processus de travail de courtier
amqrmppa	Processus de regroupement de processus
amqcrsta	Processus de travail de répondeur non à unités d'exécution
amqcrs6b	Connexion client et canal récepteur LU62
amqrrmfa	Processus de référentiel (pour les clusters)
amqpcsea	Serveur de commandes
runmqtrm	Appel d'un moniteur de déclenchement pour un serveur
runmqdlq	Appeler le gestionnaire de files d'attente de rebut
runmqchi	Processus de l'initiateur de canal
runmqlsr	Processus du programme d'écoute de canal

### Tâches associées

«Arrêt d'un gestionnaire de files d'attente», à la page 136

Vous pouvez utiliser la commande **endmqm** pour arrêter un gestionnaire de files d'attente. Cette commande permet d'arrêter un gestionnaire de files d'attente de quatre manières: un arrêt contrôlé ou mis au repos, un arrêt immédiat, un arrêt préventif et un arrêt en attente. Vous pouvez également, sous Windows et Linux, arrêter un gestionnaire de files d'attente à l'aide de la IBM MQ Explorer.

### **Multi** Redémarrage d'un gestionnaire de files d'attente

Vous pouvez utiliser la commande **strmqm** pour redémarrer un gestionnaire de files d'attente ou, sur les systèmes Windows et Linux x86-64, vous pouvez redémarrer un gestionnaire de files d'attente depuis IBM MQ Explorer.

### Pourquoi et quand exécuter cette tâche

Vous pouvez redémarrer un gestionnaire de files d'attente à l'aide de la commande **strmqm**. Pour une description de la commande **strmqm** et de ses options, voir [strmqm](#).



**Windows** **Linux** Sur les systèmes Windows et Linux x86-64, vous pouvez redémarrer un gestionnaire de files d'attente en utilisant la IBM MQ Explorer de la même manière que pour démarrer un gestionnaire de files d'attente.

## Procédure

- Pour redémarrer un gestionnaire de files d'attente à l'aide de la commande **strmqm**, entrez la commande suivie du nom du gestionnaire de files d'attente à redémarrer.

Par exemple, pour démarrer un gestionnaire de files d'attente appelé `strmqm saturn.queue.manager`, entrez la commande suivante:

```
strmqm saturn.queue.manager
```

-   Pour démarrer un gestionnaire de files d'attente à l'aide de la console IBM MQ Explorer, procédez comme suit:
  - a) Ouvrez IBM MQ Explorer.
  - b) Dans la vue Navigator, sélectionnez le gestionnaire de files d'attente.
  - c) Cliquez sur **Démarrer**.

## Résultats

Le gestionnaire de files d'attente redémarre.

Si le redémarrage du gestionnaire de files d'attente prend plus de quelques secondes, IBM MQ émet des messages d'information détaillant par intermittence la progression du démarrage.

## Affichage et modification des attributs du gestionnaire de files d'attente

La commande **DISPLAY QMGR MQSC** permet d'afficher les paramètres d'un gestionnaire de files d'attente. La commande **ALTER QMGR MQSC** permet de modifier les paramètres d'un gestionnaire de files d'attente local.

## Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

## Procédure

- Pour afficher les attributs du gestionnaire de files d'attente spécifié dans la commande **runmqsc**, utilisez la commande **DISPLAY QMGR MQSC**:

```
DISPLAY QMGR
```

L'exemple suivant illustre la sortie standard de cette commande:

```
DISPLAY QMGR
  1 : DISPLAY QMGR
AMQ8408I: Display Queue Manager details.
QMNAME(QM1)                ACCTCONO(DISABLED)
ACCTINT(1800)              ACCTMQI(OFF)
ACCTQ(OFF)                 ACTIVREC(MSG)
ACTVCONO(DISABLED)        ACTVTRC(OFF)
ADVCAP(DISABLED)          ALTDATE(2022-05-05)
ALTTIME(14.24.34)         AMQPCAP(NO)
AUTHOREV(DISABLED)        CCSID(437)
CERTLABL(ibmwebspheremqm1) CERTVPOL(ANY)
CHAD(DISABLED)            CHADEV(DISABLED)
CHAEXIT( )                CHLEV(DISABLED)
CHLAUTH(ENABLED)         CLWLDATA( )
CLWLEXIT( )              CLWLLEN(100)
CLWLMRUC(999999999)      CLWLUSEQ(LOCAL)
CMDDEV(DISABLED)         CMDLEVEL(930)
COMMANDQ(SYSTEM.ADMIN.COMMAND.QUEUE) CONFIGEV(DISABLED)
```

```

CONNAUTH(SYSTEM.DEFAULT.AUTHINFO.IDPWOS)
CRDATE(2020-12-22)
CUSTOM( )
DEFCLXQ(SCTQ)
DESCR( )
IMGINTVL(60)
IMGRCOVO(YES)
IMGSCHEM(MANUAL)
INITKEY( )
LOCALEV(DISABLED)
MARKINT(5000)
MAXMSGL(4194304)
MAXPRTY(9)
MONACLS(QMGR)
MONQ(OFF)
PERFMV(DISABLED)
PSMODE(ENABLED)
PSNPMMSG(DISCARD)
PSRTPCNT(5)
QMID(QM1_2020-12-22_15.42.49)
REPOS( )
REVDNS(ENABLED)
SCHINIT(QMGR)
SPLCAP(DISABLED)
SSLCRYP( )
SSLFIPS(NO)
SSLKEYR(C:\ProgramData\IBM\MQ\mqgrs\QM1\ssl\key)
SSLRKEYC(32767)
STATCHL(OFF)
STATMQI(OFF)
STRSTPEV(ENABLED)
SYNCPT
TRIGINT(999999999)
XRCAP(NO)
CRTIME(15.42.49)
DEADQ( )
DEFXMITQ( )
DISTL(YES)
IMGLOGLN(OFF)
IMGRCOVQ(YES)
INHIBTEV(DISABLED)
IPADDRV(IPV4)
LOGGERSV(DISABLED)
MAXHANDS(256)
MAXPROPL(NOLIMIT)
MAXUMSGS(10000)
MONCHL(OFF)
PARENT( )
PLATFORM(WINDOWS10)
PSCLUS(ENABLED)
PSNPRES(NORMAL)
PSSYNCP(IFPER)
REMOOTESV(DISABLED)
REPOSNL( )
ROUTEREC(MSG)
SCMDSERV(QMGR)
SSLCRLNL( )
SSLEV(DISABLED)
KEYRPWD( )
STATACLS(QMGR)
STATINT(1800)
STATQ(OFF)
SUITEB(NONE)
TREELIFE(1800)
VERSION(09030000)

```

**Remarque :** SYNCPT est un attribut de gestionnaire de files d'attente en lecture seule.

Le paramètre **ALL** est la valeur par défaut de la commande **DISPLAY QMGR**. Il affiche tous les attributs du gestionnaire de files d'attente. En particulier, la sortie vous indique le nom du gestionnaire de files d'attente par défaut, le nom de la file d'attente de rebut et le nom de la file d'attente de commandes.

Vous pouvez confirmer que ces files d'attente existent en entrant la commande suivante:

```
DISPLAY QUEUE (SYSTEM.*)
```

La liste des files d'attente correspondant au radical SYSTEM.\*s'affiche. Les parenthèses sont obligatoires.

- Pour modifier les attributs du gestionnaire de files d'attente spécifié dans la commande **runmqsc**, utilisez la commande **MQSC ALTER QMGR** en spécifiant les attributs et les valeurs que vous souhaitez modifier.

Par exemple, utilisez les commandes suivantes pour modifier les attributs de `jupiter.queue.manager`:

```
runmqsc jupiter.queue.manager
ALTER QMGR DEADQ (ANOTHERDLQ) INHIBTEV (ENABLED)
```

La commande **ALTER QMGR** modifie la file d'attente de rebut utilisée et active les événements d'interdiction.

Si des paramètres ne sont pas spécifiés dans la commande **ALTER QMGR**, les valeurs existantes de ces paramètres restent inchangées.

### Tâches associées

[Création de gestionnaires de files d'attente sur Multiplatforms](#)

### Référence associée

[Attributs du gestionnaire de files d'attente](#)

[runmqsc \(exécution des commandes MQSC\)](#)



## Suppression d'un gestionnaire de files d'attente

Vous pouvez supprimer un gestionnaire de files d'attente à l'aide de la commande de contrôle **dltmqm**. Sur les systèmes Windows et Linux, vous pouvez également utiliser IBM MQ Explorer pour supprimer un gestionnaire de files d'attente.

### Avant de commencer



#### Avertissement :

- La suppression d'un gestionnaire de files d'attente est une étape drastique, car vous supprimez également toutes les ressources associées au gestionnaire de files d'attente, y compris toutes les files d'attente et leurs messages, ainsi que toutes les définitions d'objet. Si vous utilisez la commande de contrôle **dltmqm**, aucune invite ne s'affiche pour vous permettre de changer d'avis. Lorsque vous appuyez sur la touche Entrée, toutes les ressources associées sont perdues.
- **Windows** Sous Windows, la suppression d'un gestionnaire de files d'attente supprime également le gestionnaire de files d'attente de la liste de démarrage automatique (décrite dans «Démarrage d'un gestionnaire de files d'attente», à la page 135). Une fois la commande terminée, un message IBM MQ queue manager ending s'affiche ; vous n'êtes pas informé que le gestionnaire de files d'attente a été supprimé.
- La suppression d'un gestionnaire de files d'attente de cluster ne le supprime pas du cluster. Pour plus d'informations, voir les remarques sur l'utilisation dans [dltmqm](#).

### Pourquoi et quand exécuter cette tâche

Vous pouvez supprimer un gestionnaire de files d'attente à l'aide de la commande de contrôle **dltmqm**. Pour une description de la commande **dltmqm** et de ses options, voir [dltmqm](#). Assurez-vous que seuls les administrateurs de confiance ont le droit d'utiliser cette commande. (Pour plus d'informations sur la sécurité, voir [Configuration de la sécurité sur AIX, Linux, and Windows](#).)

Sur les systèmes Windows et Linux (plateformes x86 et x86-64), vous pouvez également supprimer un gestionnaire de files d'attente à l'aide de la IBM MQ Explorer.

### Procédure

- Pour supprimer un gestionnaire de files d'attente à l'aide de la commande **dltmqm**, procédez comme suit:
  - a) Arrêtez le gestionnaire de files d'attente.
  - b) Entrez la commande suivante :

```
dltmqm QMB
```

**Remarque :** Vous devez utiliser la commande **dltmqm** à partir de l'installation associée au gestionnaire de files d'attente que vous utilisez. L'installation à laquelle un gestionnaire de files d'attente est associé peut être identifiée à l'aide de la commande `dspmqr -o installation`.

- Pour supprimer un gestionnaire de files d'attente à l'aide de la console IBM MQ Explorer, procédez comme suit:
  - a) Ouvrez IBM MQ Explorer.
  - b) Dans la vue Navigator, sélectionnez le gestionnaire de files d'attente.
  - c) Si le gestionnaire de files d'attente n'est pas arrêté, arrêtez-le.

Pour arrêter le gestionnaire de files d'attente, cliquez dessus avec le bouton droit de la souris, puis cliquez sur **Arrêter**.

d) Supprimez le gestionnaire de files d'attente.

Pour supprimer le gestionnaire de files d'attente, cliquez dessus avec le bouton droit de la souris, puis cliquez sur **Supprimer**.

## Résultats

Le gestionnaire de files d'attente est supprimé.

## Arrêt des canaux MQI

Lorsque vous émettez une commande STOP CHANNEL sur un canal de connexion serveur, vous pouvez choisir la méthode à utiliser pour arrêter le canal de connexion client. Cela signifie qu'un canal client établissant un appel d'attente MQGET peut être contrôlé et que vous pouvez décider comment et quand arrêter le canal.

La commande STOP CHANNEL peut être exécutée selon trois méthodes, en indiquant comment le canal doit être arrêté :

### Quiesce (mise au repos)

Arrête le canal une fois que tous les messages en cours ont été traités.

Si le partage de conversations est activé, le IBM MQ MQI client prend connaissance de la demande d'arrêt dans les délais ; cette durée dépend de la vitesse du réseau. L'application client prend connaissance de la demande d'arrêt suite à l'émission d'un appel ultérieur à IBM MQ.

### Force

Arrête le canal immédiatement.

### Terminer

Arrête le canal immédiatement. Si le canal est en cours d'exécution sous la forme d'un processus, il peut mettre fin au processus du canal ou, si le canal est en cours d'exécution sous la forme d'une unité d'exécution, à son unité d'exécution.

Il s'agit d'un processus à plusieurs phases. Si le mode terminate est utilisé, une tentative d'arrêt du canal de connexion serveur est effectuée, d'abord en mode quiesce, puis en mode force et, si nécessaire, en mode terminate. Le client peut recevoir des codes retour différents pendant les différentes phases d'arrêt. Si le processus ou l'unité d'exécution est arrêtée, le client reçoit une erreur de communication.

Les codes retour renvoyés à l'application varient selon l'appel MQI émis et la commande STOP CHANNEL exécutée. Le client reçoit un code retour MQRC\_CONNECTION\_QUIESCING ou MQRC\_CONNECTION\_BROKEN. Si un client détecte MQRC\_CONNECTION\_QUIESCING, il doit tenter de terminer la transaction en cours et de s'arrêter. Cela n'est pas possible avec MQRC\_CONNECTION\_BROKEN. Si le client ne termine pas la transaction et ne s'arrête pas suffisamment vite, il reçoit le code retour CONNECTION\_BROKEN au bout de quelques secondes. Une commande STOP CHANNEL avec MODE(FORCE) ou MODE(TERMINATE) est plus susceptible de générer un code CONNECTION\_BROKEN qu'avec MODE(QUIESCE).

### Concepts associés

[Canaux](#)

## Utilisation des files d'attente locales

Cette section contient des exemples de certaines commandes MQSC que vous pouvez utiliser pour gérer les files d'attente locales, de modèle et d'alias.

Pour plus d'informations sur ces commandes, voir [Commandes MQSC](#).

### Référence associée

[Restrictions de dénomination pour les files d'attente](#)

[Restrictions de dénomination pour d'autres objets](#)

## Définition d'une file d'attente locale avec DEFINE QLOCAL

Pour une application, le gestionnaire de files d'attente local est le gestionnaire de files d'attente auquel l'application est connectée. Les files d'attente gérées par le gestionnaire de files d'attente local sont dites locales pour ce gestionnaire de files d'attente. Vous utilisez la commande MQSC **DEFINE QLOCAL** pour créer une file d'attente locale.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Pourquoi et quand exécuter cette tâche

Vous utilisez la commande MQSC **DEFINE QLOCAL** pour créer une file d'attente locale. Vous pouvez également utiliser la valeur par défaut définie dans la définition de file d'attente locale par défaut ou modifier les caractéristiques de la file d'attente locale par défaut.

**Remarque :** La file d'attente locale par défaut est nommée SYSTEM.DEFAULT.LOCAL.QUEUE et il est créé sur l'installation du système.

### Procédure

- Pour créer une file d'attente locale, entrez la commande **DEFINE QLOCAL** comme illustré dans l'exemple suivant.

Dans cet exemple, la commande **DEFINE QLOCAL** définit une file d'attente appelée ORANGE.LOCAL.QUEUE avec les caractéristiques suivantes:

- Il est activé pour les requêtes, activé pour les insertions et fonctionne en fonction de l'ordre de priorité.
- Il s'agit d'une file d'attente *normale* ; il ne s'agit pas d'une file d'attente d'initialisation ou de transmission et elle ne génère pas de messages de déclenchement.
- La longueur maximale de la file d'attente est de 5000 messages ; la longueur maximale est de 4194304 octets.

```
DEFINE QLOCAL(ORANGE.LOCAL.QUEUE) +
  DESCR('Queue for messages from other systems') +
  PUT(ENABLED) +
  GET(ENABLED) +
  NOTRIGGER +
  MSGDLVSQ(PRIORITY) +
  MAXDEPTH(5000) +
  MAXMSGL(4194304) +
  USAGE(NORMAL)
```

#### Remarques :

1. A l'exception de la valeur de la description, toutes les valeurs d'attribut affichées dans l'exemple sont les valeurs par défaut. Ces exemples sont inclus à des fins d'illustration. Vous pouvez les omettre si vous êtes sûr que les valeurs par défaut correspondent à celles que vous souhaitez ou qu'elles n'ont pas été modifiées. Voir aussi [«Affichage des attributs d'objet par défaut avec DISPLAY QUEUE»](#), à la page 148.
2. **USAGE(NORMAL)** indique que cette file d'attente n'est pas une file d'attente de transmission.
3. Si vous disposez déjà d'une file d'attente locale sur le même gestionnaire de files d'attente avec le nom ORANGE.LOCAL.QUEUE, cette commande échoue. Utilisez l'attribut **REPLACE** si vous souhaitez remplacer la définition existante d'une file d'attente, mais voir aussi [«Modification des attributs de file d'attente locale avec ALTER QLOCAL ou DEFINE QLOCAL»](#), à la page 149.

## Référence associée

[DEFINE QLOCAL](#)

## Affichage des attributs d'objet par défaut avec DISPLAY QUEUE

Vous pouvez utiliser la commande **DISPLAY QUEUE** MQSC pour afficher les attributs qui ont été extraits de l'objet par défaut lorsqu'un objet IBM MQ a été défini.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Pourquoi et quand exécuter cette tâche

Lorsque vous définissez un objet IBM MQ , il prend tous les attributs que vous ne spécifiez pas à partir de l'objet par défaut. Par exemple, lorsque vous définissez une file d'attente locale, celle-ci hérite de tous les attributs que vous omettez dans la définition de la file d'attente locale par défaut, appelée SYSTEM.DEFAULT.LOCAL.QUEUE. Vous pouvez utiliser la commande **DISPLAY QUEUE** pour voir exactement quels sont ces attributs.

### Procédure

- Pour afficher les attributs d'objet par défaut d'une file d'attente locale, utilisez la commande suivante:

```
DISPLAY QUEUE (SYSTEM.DEFAULT.LOCAL.QUEUE)
```

La syntaxe de la commande **DISPLAY** est différente de celle de la commande **DEFINE** correspondante. Dans la commande **DISPLAY** , vous pouvez donner uniquement le nom de la file d'attente, tandis que dans la commande **DEFINE** , vous devez spécifier le type de la file d'attente, c'est-à-dire QLOCAL, QALIAS, QMODEL ou QREMOTE.

Vous pouvez afficher les attributs de manière sélective en les spécifiant individuellement. Exemple :

```
DISPLAY QUEUE (ORANGE.LOCAL.QUEUE) +  
MAXDEPTH +  
MAXMSGL +  
CURDEPTH;
```

Cette commande affiche les trois attributs spécifiés comme suit:

```
AMQ8409: Display Queue details.  
QUEUE(ORANGE.LOCAL.QUEUE)      TYPE(QLOCAL)  
CURDEPTH(0)                     MAXDEPTH(5000)  
MAXMSGL(4194304)
```

**CURDEPTH** est la longueur de la file d'attente en cours, c'est-à-dire le nombre de messages dans la file d'attente. Il s'agit d'un attribut utile à afficher, car en surveillant la longueur de la file d'attente, vous pouvez vous assurer que la file d'attente n'est pas saturée.

## Référence associée

[DISPLAY QUEUE](#)

[DEFINE QUEUES](#)

## Copie d'une définition de file d'attente locale avec DEFINE QLOCAL

Vous pouvez copier une définition de file d'attente à l'aide de l'attribut **LIKE** de la commande **DEFINE QLOCAL** MQSC.

## Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

## Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser la commande **DEFINE** avec l'attribut **LIKE** pour créer une file d'attente avec les mêmes attributs que la file d'attente spécifiée, plutôt que ceux de la file d'attente locale par défaut du système. Vous pouvez également utiliser cette forme de la commande **DEFINE** pour copier une définition de file d'attente, mais remplacer une ou plusieurs modifications apportées aux attributs de l'original.

### Remarques :

1. Lorsque vous utilisez l'attribut **LIKE** dans une commande **DEFINE** , vous copiez uniquement les attributs de file d'attente. Vous ne copiez pas les messages de la file d'attente.
2. Si vous définissez une file d'attente locale, sans spécifier **LIKE**, elle est identique à:

```
DEFINE LIKE(SYSTEM.DEFAULT.LOCAL.QUEUE)
```

## Procédure

- Pour créer une file d'attente avec les mêmes attributs que la file d'attente spécifiée, plutôt que ceux de la file d'attente locale par défaut du système, entrez la commande **DEFINE** comme illustré dans l'exemple suivant.

Entrez le nom de la file d'attente à copier exactement tel qu'il a été entré lors de la création de la file d'attente. Si le nom contient des caractères minuscules, placez-le entre apostrophes.

Cet exemple crée une file d'attente avec les mêmes attributs que la file d'attente ORANGE.LOCAL.QUEUE, plutôt que celles de la file d'attente locale par défaut du système:

```
DEFINE QLOCAL (MAGENTA.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE)
```

- Pour copier une définition de file d'attente, mais remplacer une ou plusieurs modifications apportées aux attributs de l'original, entrez la commande **DEFINE** comme illustré dans l'exemple suivant. Cette commande copie les attributs de la file d'attente ORANGE.LOCAL.QUEUE dans la file d'attente THIRD.QUEUE, mais indique que la longueur maximale des messages dans la nouvelle file d'attente doit être de 1024 octets au lieu de 4194304:

```
DEFINE QLOCAL (THIRD.QUEUE) +  
LIKE (ORANGE.LOCAL.QUEUE) +  
MAXMSGL(1024);
```

### Référence associée

[DEFINE QUEUES](#)

## Modification des attributs de file d'attente locale avec ALTER QLOCAL ou DEFINE QLOCAL

Vous pouvez modifier les attributs de file d'attente de deux manières, à l'aide de la commande **ALTER QLOCAL** ou **DEFINE QLOCAL** MQSC avec l'attribut **REPLACE** .

## Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

## Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'attribut **REPLACE** de la commande **ALTER** et de la commande **DEFINE** pour remplacer une définition existante par la nouvelle définition spécifiée. La différence entre l'utilisation de **ALTER** et de **DEFINE** est que **ALTER** avec **REPLACE** ne modifie pas les paramètres non spécifiés, mais **DEFINE** avec **REPLACE** définit tous les paramètres.

## Procédure

- Pour modifier les attributs de file d'attente, utilisez la commande **ALTER** ou la commande **DEFINE**, comme illustré dans les exemples suivants.

Dans ces exemples, la longueur maximale des messages dans la file d'attente ORANGE.LOCAL.QUEUE est réduit à 10 000 octets.

- A l'aide de la commande **ALTER** :

```
ALTER QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000)
```

Cette commande modifie un attribut unique, celui de la longueur maximale de message ; tous les autres attributs restent identiques.

- Utilisation de la commande **DEFINE** avec l'option **REPLACE**, par exemple:

```
DEFINE QLOCAL (ORANGE.LOCAL.QUEUE) MAXMSGL(10000) REPLACE
```

Cette commande modifie non seulement la longueur maximale des messages, mais également tous les autres attributs auxquels sont affectées leurs valeurs par défaut. Ainsi, par exemple, si la file d'attente a été précédemment interdite, elle est modifiée pour être activée car l'insertion activée est la valeur par défaut, comme indiqué par la file d'attente SYSTEM.DEFAULT.LOCAL.QUEUE.

Si vous réduisez la longueur maximale des messages dans une file d'attente existante, les messages existants ne sont pas affectés. Toutefois, tout nouveau message doit répondre aux nouveaux critères.

### Référence associée

[Files d'attente ALTER](#)

[ALTER QLOCAL](#)

[DEFINE QUEUES](#)

[DEFINE QLOCAL](#)

## Effacement d'une file d'attente locale avec CLEAR QLOCAL

Vous pouvez utiliser la commande **CLEAR QLOCAL MQSC** pour effacer une file d'attente locale.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d'IBM MQ à l'aide de commandes MQSC](#).

Vous ne pouvez pas effacer une file d'attente si:

- Des messages non validés ont été placés dans la file d'attente sous le point de synchronisation.
- La file d'attente est ouverte par une application.

## Pourquoi et quand exécuter cette tâche

Si vous souhaitez effacer une file d'attente locale à l'aide de la commande **CLEAR QLOCAL**, le nom de la file d'attente doit être défini dans le gestionnaire de files d'attente local.

**Remarque :** Aucune invite ne vous permet de changer d'avis. Une fois que vous appuyez sur la touche Entrée, les messages sont perdus.

## Procédure

Pour effacer les messages d'une file d'attente locale, utilisez **CLEAR QLOCAL** comme illustré dans l'exemple suivant.

Dans cet exemple, tous les messages sont supprimés d'une file d'attente locale appelée MAGENTA.QUEUE:

```
CLEAR QLOCAL (MAGENTA.QUEUE)
```

### Référence associée

[QLOCAL CLEAR](#)

## Suppression d'une file d'attente locale avec DELETE QLOCAL

Vous pouvez utiliser la commande **DELETE QLOCAL** MQSC pour supprimer une file d'attente locale.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

Une file d'attente ne peut pas être supprimée si elle contient des messages non validés.

Si une file d'attente comporte un ou plusieurs messages validés et aucun message non validé, elle ne peut être supprimée que si vous spécifiez l'option **PURGE**. La suppression se poursuit même s'il existe des messages validés dans la file d'attente nommée et que ces messages sont également purgés.

La spécification de **NOPURGE** à la place de **PURGE** garantit que la file d'attente n'est pas supprimée si elle contient des messages validés.

## Procédure

- Pour supprimer une file d'attente locale, utilisez la commande **DELETE QLOCAL** comme illustré dans l'exemple suivant.

Cet exemple supprime la file d'attente PINK.QUEUE s'il n'y a pas de messages validés dans la file d'attente:

```
DELETE QLOCAL (PINK.QUEUE) NOPURGE
```

Cet exemple supprime la file d'attente PINK.QUEUE même s'il existe des messages validés dans la file d'attente:

```
DELETE QLOCAL (PINK.QUEUE) PURGE
```

### Référence associée

[SUPPRIMER QLOCAL](#)

## Exploration des files d'attente avec l'exemple de programme

IBM MQ fournit un exemple de navigateur de files d'attente que vous pouvez utiliser pour examiner le contenu des messages d'une file d'attente.





```
MsgSeqNumber : '1'  
Offset       : '0'  
MsgFlags    : '0'  
OriginalLength : '104'
```

\*\*\*\* Message \*\*\*\*

length - 104 bytes

```
00000000: 0700 0000 2400 0000 0100 0000 2C00 0000 | .....->.....'  
00000010: 0100 0000 0100 0000 0100 0000 AE08 0000 | .....'  
00000020: 0100 0000 0400 0000 4400 0000 DF07 0000 | .....D.....'  
00000030: 0000 0000 3000 0000 7361 7475 726E 2E71 | .....0...saturn.q'  
00000040: 7565 7565 2E6D 616E 6167 6572 2020 2020 | ueue.manager '  
00000050: 2020 2020 2020 2020 2020 2020 2020 2020 | .....'  
00000060: 2020 2020 2020 2020 | .....
```

```
No more messages  
MQCLOSE  
MQDISC
```

## Référence associée

[Exemple de programme de navigateur](#)

## Activation des files d'attente volumineuses

IBM MQ prend en charge les files d'attente supérieures à 2 To.

**Windows** Sur les systèmes Windows , la prise en charge des fichiers volumineux est disponible sans activation supplémentaire.

**Linux** **AIX** Sur les systèmes AIX and Linux , vous devez activer explicitement la prise en charge des fichiers volumineux avant de pouvoir créer des fichiers de file d'attente de plusieurs gigaoctets ou téraoctets. Consultez la documentation de votre système d'exploitation pour plus d'informations sur la procédure à suivre.

Certains utilitaires, tels que tar, ne peuvent pas gérer des fichiers de plusieurs gigaoctets ou téraoctets. Avant d'activer la prise en charge des fichiers volumineux, consultez la documentation de votre système d'exploitation pour plus d'informations sur les restrictions relatives aux utilitaires que vous utilisez.

Pour plus d'informations sur la planification de la quantité de stockage dont vous avez besoin pour les files d'attente, voir [MQ Performance documents](#) pour les rapports de performances spécifiques à la plateforme.

Vous pouvez contrôler la taille des fichiers de file d'attente à l'aide d'un nouvel attribut sur les files d'attente locales et modèles. Pour plus d'informations, voir [«Modification des fichiers de file d'attente IBM MQ»](#), à la page 153.

## Multi Modification des fichiers de file d'attente IBM MQ

Vous pouvez contrôler la taille des fichiers de file d'attente à l'aide d'un attribut sur les files d'attente locales et modèles. Vous pouvez afficher la taille actuelle d'un fichier de file d'attente et la taille maximale qu'il peut atteindre (en fonction de la taille de bloc actuellement utilisée dans ce fichier), à l'aide de deux attributs d'état de file d'attente.

## Attribut utilisé pour modifier les fichiers de file d'attente

L'attribut sur les files d'attente locales et modèles est:

### MAXFSIZE

Indique la taille maximale du fichier de file d'attente utilisé par la file d'attente, en mégaoctets.

Vous pouvez définir ou afficher la valeur de cet attribut à l'aide des commandes MQSC, IBM MQ Explorer ou administrative REST API. Vous pouvez également afficher la valeur de cet attribut dans IBM MQ Console.

Pour plus d'informations, voir [MAXFSIZE](#) et «[Modification de la taille d'un fichier de file d'attente IBM MQ](#)», à la page 154 .

L'équivalent PCF de cet attribut est **MQIA\_MAX\_Q\_FILE\_SIZE**. Voir [Modification, copie et création de file d'attente](#).

Les deux attributs du statut de la file d'attente sont les suivants:

#### **CURFSIZE**

Affiche la taille actuelle du fichier de file d'attente en mégaoctets, arrondie au mégaoctet le plus proche.

Vous pouvez définir ou afficher la valeur de cet attribut à l'aide des commandes MQSC, IBM MQ Explorer ou administrative REST API.

Pour plus d'informations, voir [CURFSIZE](#) .

L'équivalent PCF de cet attribut est **MQIA\_CUR\_Q\_FILE\_SIZE**. Voir [Inquire Queue](#) et [Inquire Queue \(response\)](#).

#### **CURMAXFS**

Indique la taille maximale actuelle que le fichier de file d'attente peut atteindre, arrondie au mégaoctet le plus proche, compte tenu de la taille de bloc en cours d'utilisation dans une file d'attente.

Vous pouvez définir ou afficher la valeur de cet attribut à l'aide des commandes MQSC, IBM MQ Explorer ou administrative REST API.

Pour plus d'informations, voir [CURMAXFS](#) .

L'équivalent PCF de cet attribut est **MQIA\_CUR\_MAX\_FILE\_SIZE**. Voir [Inquire Queue](#) et [Inquire Queue \(response\)](#).

## **Taille et granularité des blocs**

Les fichiers de file d'attente sont divisés en segments appelés blocs. Pour augmenter la taille maximale d'un fichier de file d'attente, la taille de bloc ou la granularité de la file d'attente doit être modifiée par le gestionnaire de files d'attente.

Si une file d'attente nouvellement définie est créée avec une valeur **MAXFSIZE** élevée, elle est créée avec une taille de bloc appropriée. Toutefois, si la valeur **MAXFSIZE** d'une file d'attente existante est augmentée, par exemple à l'aide de la commande **ALTER QLOCAL** MQSC, il peut être nécessaire d'autoriser le vidage de la file d'attente pour que le gestionnaire de files d'attente reconfigure la file d'attente.

Pour plus d'informations, voir «[Calcul de la quantité de données qu'un fichier de file d'attente IBM MQ peut stocker](#)», à la page 156.



**Avertissement :** Certains systèmes de fichiers et systèmes d'exploitation ont des limites sur la taille de l'ensemble du système de fichiers ou sur la taille d'un fichier individuel. Vous devez vérifier les limites sur les systèmes utilisés par votre entreprise.

#### **Référence associée**

[MODIFIER LES FILES D'ATTENTE](#)

[DISPLAY QUEUE](#)

[STATUT DE LA FILE D'ATTENTE D'AFFICHAGE](#)



### **Modification de la taille d'un fichier de file d'attente IBM MQ**

Vous pouvez augmenter ou diminuer la taille maximale d'un fichier de file d'attente.

## **Avant de commencer**

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC:

- **ALW** Sous AIX, Linux, and Windows, vous émettez des commandes MQSC à partir d'une invite de commande **runmqsc** . Voir [Exécution interactive de commandes MQSC sous runmqsc](#) et [Exécution de commandes MQSC à partir de fichiers texte sous runmqsc](#).
- **IBM i** Sous IBM i, vous créez une liste de commandes dans un fichier script, puis vous exécutez le fichier à l'aide de la commande **STRMQMMQSC** . Voir [Administration à l'aide de commandes MQSC sous IBM i](#).

Avant de définir une nouvelle taille pour un fichier de file d'attente, utilisez la commande MQSC [DISPLAY QLOCAL](#) pour afficher la taille du fichier de file d'attente que vous souhaitez modifier. Par exemple, exécutez la commande suivante :

```
DISPLAY QLOCAL (SYSTEM.DEFAULT.LOCAL.QUEUE) MAXFSIZE
```

La sortie suivante est générée :

```
AMQ8409I: Display queue details
          QUEUE(SYSTEM.DEFAULT.LOCAL.QUEUE)          TYPE(QLOCAL)
          MAXFSIZE(DEFAULT)
```

qui indique que la taille maximale du fichier de file d'attente est la valeur par défaut de 2 088 960 Mo.

## Pourquoi et quand exécuter cette tâche

Les procédures suivantes vous montrent comment:

- Réduisez la taille maximale qu'un fichier de file d'attente peut atteindre.
- Augmentez la taille maximale qu'un fichier de file d'attente peut atteindre.



**Avertissement :** Vous devez être prudent lorsque vous augmentez la taille des fichiers de file d'attente sans tenir compte de la manière dont vos applications sont écrites et de l'impact possible sur les performances. L'accès aux messages au hasard dans un fichier de file d'attente très volumineux peut être très lent.

Si vous envisagez d'augmenter la taille maximale d'un fichier de file d'attente au-delà de la valeur par défaut, vous devez être prudent lorsque vous utilisez des sélecteurs de message tels que des ID de corrélation et des chaînes de sélecteur IBM MQ classes for JMS **JM 3.0** ou [IBM MQ classes for Jakarta Messaging](#) . Les fichiers de file d'attente plus volumineux sont mieux adaptés à l'accès premier entré, premier sorti à la file d'attente.

La présence de très grandes quantités de données dans des fichiers de file d'attente individuels ne doit être effectuée que sur les gestionnaires de files d'attente configurés pour la journalisation avec réutilisation automatique des journaux ou lorsque l'imagerie du support n'a pas été activée pour la file d'attente individuelle.

Vous ne devez pas limiter la taille des files d'attente SYSTEM car cela pourrait affecter le fonctionnement du gestionnaire de files d'attente.

## Procédure

1. Réduction de la taille maximale du fichier de file d'attente
  - a) Emettez la commande MQSC suivante pour créer un fichier local appelé SMALLQUEUE, d'une taille de 500 gigaoctets:

```
DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
  2 : DEFINE QLOCAL(SMALLQUEUE) MAXFSIZE(512000)
AMQ8006I: IBM MQ queue created
```

et vous recevez le message AMQ8006I:

**Remarque :** Si vous configurez une file d'attente avec une valeur inférieure à la quantité de données figurant déjà dans le fichier, les nouveaux messages ne peuvent pas être placés dans la file d'attente.

Si une application tente d'insérer un message dans un fichier de file d'attente qui ne dispose pas de suffisamment d'espace, elle reçoit le code retour MQRC\_Q\_SPACE\_NOT\_AVAILABLE. Lorsqu'un nombre suffisant de messages est lu de façon destructive à partir de la file d'attente, les applications peuvent commencer à insérer de nouveaux messages dans la file d'attente.

2. Augmentez la taille maximale du fichier de file d'attente.

a) Emettez la commande MQSC suivante pour créer un fichier local appelé LARGEQUEUE, d'une taille de 5 téraoctets:

```
DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
3 : DEFINE QLOCAL(LARGEQUEUE) MAXFSIZE(5242880)
AMQ8006I: IBM MQ queue created
```

## Multi **Calcul de la quantité de données qu'un fichier de file d'attente IBM MQ peut stocker**

La quantité de données pouvant être stockées dans une file d'attente est limitée par la taille des blocs individuels dans lesquels la file d'attente est divisée. Utilisez les commandes MQSC pour confirmer la taille de bloc et la granularité, et vérifiez la taille d'un fichier de file d'attente.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC:

- ▶ **ALW** Sous AIX, Linux, and Windows, vous émettez des commandes MQSC à partir d'une invite de commande **runmqsc**. Voir [Exécution interactive de commandes MQSC sous runmqsc](#) et [Exécution de commandes MQSC à partir de fichiers texte sous runmqsc](#).
- ▶ **IBM i** Sous IBM i, vous créez une liste de commandes dans un fichier script, puis vous exécutez le fichier à l'aide de la commande **STRMQMMQSC**. Voir [Administration à l'aide de commandes MQSC sous IBM i](#).

### Procédure

- Confirmez la taille et la granularité des blocs.

La taille de bloc par défaut est de 512 octets. Pour prendre en charge les fichiers de file d'attente dont la taille est supérieure à deux téraoctets, le gestionnaire de files d'attente doit augmenter la taille de bloc.

La taille de bloc est automatiquement calculée lorsque vous configurez **MAXFSIZE** pour une file d'attente, mais la taille de bloc révisée ne peut pas être appliquée à la file d'attente si celle-ci contient déjà des messages. Une fois qu'une file d'attente est vide, le gestionnaire de files d'attente modifie automatiquement la taille de bloc pour prendre en charge le **MAXFSIZE** configuré.

La commande **DISPLAY QSTATUS** possède un nouvel attribut, **CURMAXFS**, qui vous permet de confirmer qu'une file d'attente a été modifiée pour utiliser une nouvelle taille de bloc.

Dans l'exemple suivant, la valeur **CURMAXFS** de 4177920 confirme que le fichier de file d'attente peut actuellement atteindre une taille d'environ quatre téraoctets. Si la valeur de **MAXFSIZE** configurée sur la file d'attente est supérieure à la valeur de **CURMAXFS**, le gestionnaire de files d'attente attend toujours que la file d'attente soit vidée avant de reconfigurer la taille de bloc du fichier de file d'attente.

```
DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
2 : DISPLAY QSTATUS(LARGEQUEUE) CURMAXFS
AMQ8450I: Display queue status details
```

```
QUEUE(LARGEQUEUE)
CURMAXFS(4177920)
```

```
TYPE(Queue)
CURDEPTH(100000)
```

- Vérifiez la taille d'un fichier de file d'attente.

Vous pouvez afficher la taille actuelle d'un fichier de file d'attente sur le disque, en mégaoctets, à l'aide de l'attribut `CURFSIZE` de la commande **DISPLAY QSTATUS**. Cela peut être utile sur les plateformes telles que IBM MQ Appliance, où il n'est pas possible d'accéder directement au système de fichiers.

```
DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
1 : DISPLAY QSTATUS(SMALLQUEUE) CURFSIZE
AMQ8450I: Display queue status details
QUEUE(SMALLQUEUE) TYPE(Queue)
CURDEPTH(4024) CURFSIZE(10)
```

**Remarque :** Lorsque des messages sont supprimés d'une file d'attente, l'attribut **CURFSIZE** ne diminue pas immédiatement.

Généralement, l'espace dans un fichier de file d'attente est libéré uniquement lorsqu'aucune application n'a la file d'attente ouverte et qu'aucun message en attente de validation n'est stocké dans la file d'attente. Toute troncature ou compression nécessaire d'un fichier de file d'attente chargé par le gestionnaire de files d'attente se produit lors du point de contrôle, de l'arrêt du gestionnaire de files d'attente ou de l'enregistrement d'une image de support de la file d'attente.

### Référence associée

[MODIFIER LES FILES D'ATTENTE](#)

[STATUT DE LA FILE D'ATTENTE D'AFFICHAGE](#)

## Utilisation des files d'attente éloignées

Une file d'attente éloignée est une définition sur un gestionnaire de files d'attente local qui fait référence à une file d'attente sur un gestionnaire de files d'attente éloignées. Vous n'avez pas besoin de définir une file d'attente éloignée à partir d'une position locale, mais si vous le faites, les applications peuvent faire référence à la file d'attente éloignée par son nom défini localement au lieu d'avoir à spécifier un nom qualifié par l'ID du gestionnaire de files d'attente sur lequel se trouve la file d'attente éloignée.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d'IBM MQ à l'aide de commandes MQSC](#).

### Pourquoi et quand exécuter cette tâche

Une application se connecte à un gestionnaire de files d'attente local, puis émet un appel MQOPEN. Dans l'appel ouvert, le nom de file d'attente spécifié est celui d'une définition de file d'attente éloignée sur le gestionnaire de files d'attente local. La définition de file d'attente éloignée fournit les noms de la file d'attente cible, du gestionnaire de files d'attente cible et, le cas échéant, d'une file d'attente de transmission. Pour placer un message dans la file d'attente éloignée, l'application émet un appel MQPUT en spécifiant le descripteur renvoyé par l'appel MQOPEN. Le gestionnaire de files d'attente utilise le nom de la file d'attente éloignée et le nom du gestionnaire de files d'attente éloignées dans un en-tête de transmission au début du message. Ces informations sont utilisées pour acheminer le message vers sa destination correcte dans le réseau.

En tant qu'administrateur, vous pouvez contrôler la destination du message en modifiant la définition de la file d'attente éloignée.

### Procédure

- Placez un message dans une file d'attente appartenant à un gestionnaire de files d'attente éloignées.

L'application se connecte à un gestionnaire de files d'attente, par exemple, saturn.queue.manager. La file d'attente cible appartient à un autre gestionnaire de files d'attente.

Dans l'appel MQOPEN , l'application spécifie les zones suivantes:

Zone Valeur	Description
<i>ObjectName</i> CYAN.REMOTE.QUEUE	Indique le nom local de l'objet file d'attente éloignée. Définit la file d'attente cible et le gestionnaire de files d'attente cible.
<i>ObjectType</i> (File d'attente)	Identifie cet objet en tant que file d'attente.
<i>ObjectQmgrName</i> Vide ou saturn.queue.manager	Cette zone est facultative.  Si cette zone est vide, le nom du gestionnaire de files d'attente local est pris en compte. (Il s'agit du gestionnaire de files d'attente sur lequel la définition de file d'attente éloignée existe.)

Après cela, l'application émet un appel MQPUT pour insérer un message dans cette file d'attente.

Sur le gestionnaire de files d'attente local, vous pouvez créer une définition locale d'une file d'attente éloignée à l'aide des commandes MQSC suivantes:

```
DEFINE QREMOTE (CYAN.REMOTE.QUEUE) +
DESCR ('Queue for auto insurance requests from the branches') +
RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE) +
RQMNAME (jupiter.queue.manager) +
XMITQ (INQUOTE.XMIT.QUEUE)
```

où :

**QREMOTE (CYAN.REMOTE.QUEUE)**

Indique le nom local de l'objet file d'attente éloignée. Il s'agit du nom que les applications connectées à ce gestionnaire de files d'attente doivent spécifier dans l'appel MQOPEN pour ouvrir la file d'attente AUTOMOBILE.INSURANCE.QUOTE.QUEUE sur le gestionnaire de files d'attente éloignées jupiter.queue.manager.

**DESCR ('Queue for auto insurance requests from the branches')**

Fournit un texte supplémentaire qui décrit l'utilisation de la file d'attente.

**RNAME (AUTOMOBILE.INSURANCE.QUOTE.QUEUE)**

Indique le nom de la file d'attente cible sur le gestionnaire de files d'attente éloignées. Il s'agit de la file d'attente cible réelle pour les messages envoyés par les applications qui spécifient le nom de file d'attente CYAN.REMOTE.QUEUE. File d'attente AUTOMOBILE.INSURANCE.QUOTE.QUEUE doit être définie en tant que file d'attente locale sur le gestionnaire de files d'attente éloignées.

**RQMNAME (jupiter.queue.manager)**

Indique le nom du gestionnaire de files d'attente éloignées qui possède la file d'attente cible AUTOMOBILE.INSURANCE.QUOTE.QUEUE.

**XMITQ (INQUOTE.XMIT.QUEUE)**

Définit le nom de la file d'attente de transmission. Cette option est facultative ; si le nom d'une file d'attente de transmission n'est pas spécifié, une file d'attente portant le même nom que le gestionnaire de files d'attente éloignées est utilisée.

Dans les deux cas, la file d'attente de transmission appropriée doit être définie en tant que file d'attente locale avec un attribut **Usage** spécifiant qu'il s'agit d'une file d'attente de transmission (USAGE (XMITQ)) dans les commandes MQSC).

- Insertion de messages dans une file d'attente éloignée (méthode alternative).

L'utilisation d'une définition locale d'une file d'attente éloignée n'est pas le seul moyen d'insérer des messages dans une file d'attente éloignée. Les applications peuvent spécifier le nom complet de la file d'attente, y compris le nom du gestionnaire de files d'attente éloignées, dans le cadre de l'appel MQOPEN . Dans ce cas, vous n'avez pas besoin d'une définition locale d'une file d'attente éloignée. Toutefois, cela signifie que les applications doivent connaître le nom du gestionnaire de files d'attente éloignées ou y avoir accès lors de l'exécution.

- Utilisez d'autres commandes avec des files d'attente éloignées.

Vous pouvez utiliser des commandes MQSC pour afficher ou modifier les attributs d'un objet de file d'attente éloignée, ou vous pouvez supprimer l'objet de file d'attente éloignée. Exemple :

- Pour afficher les attributs de la file d'attente éloignée:

```
DISPLAY QUEUE (CYAN.REMOTE.QUEUE)
```

- Pour modifier la file d'attente éloignée afin d'activer les insertions. Cela n'affecte pas la file d'attente cible, mais uniquement les applications qui spécifient cette file d'attente éloignée:

```
ALTER QREMOTE (CYAN.REMOTE.QUEUE) PUT(ENABLED)
```

- Permet de supprimer cette file d'attente éloignée. Cela n'affecte pas la file d'attente cible, mais uniquement sa définition locale:

```
DELETE QREMOTE (CYAN.REMOTE.QUEUE)
```

**Remarque :** Lorsque vous supprimez une file d'attente éloignée, vous supprimez uniquement la représentation locale de la file d'attente éloignée. Vous ne supprimez pas la file d'attente éloignée elle-même ou les messages qu'elle contient.

## Les définitions de file d'attente éloignée peuvent être utilisées comme alias

En plus de localiser une file d'attente sur un autre gestionnaire de files d'attente, vous pouvez également utiliser une définition locale d'une file d'attente éloignée pour les alias de gestionnaire de files d'attente et les alias de file d'attente de réponse. Les deux types d'alias sont résolus via la définition locale d'une file d'attente éloignée. Vous devez configurer les canaux appropriés pour que le message arrive à sa destination.

### Alias de gestionnaire de files d'attente

Un alias est le processus par lequel le nom du gestionnaire de files d'attente cible, tel que spécifié dans un message, est modifié par un gestionnaire de files d'attente sur la route des messages. Les alias de gestionnaire de files d'attente sont importants car vous pouvez les utiliser pour contrôler la destination des messages au sein d'un réseau de gestionnaires de files d'attente.

Pour ce faire, modifiez la définition de file d'attente éloignée sur le gestionnaire de files d'attente au point de contrôle. L'application émettrice n'est pas consciente que le nom de gestionnaire de files d'attente spécifié est un alias.

Pour plus d'informations sur les alias de gestionnaire de files d'attente, voir [Que sont les alias?](#)

### Alias de file d'attente de réponse

En option, une application peut spécifier le nom d'une file d'attente de réponse lorsqu'elle place un *message de demande* dans une file d'attente.

Si l'application qui traite le message extrait le nom de la file d'attente de réponse, elle sait où envoyer le *message de réponse*, si nécessaire.

Un alias de file d'attente de réponse est le processus par lequel une file d'attente de réponse, telle que spécifiée dans un message de demande, est modifiée par un gestionnaire de files d'attente sur la route

des messages. L'application émettrice ne sait pas que le nom de file d'attente de réponse indiqué est un alias.

Un alias de file d'attente de réponse vous permet de modifier le nom de la file d'attente de réponse et éventuellement de son gestionnaire de files d'attente. Cela vous permet de contrôler la route utilisée pour les messages de réponse.

Pour plus d'informations sur les messages de demande, les messages de réponse et les files d'attente de réponse, voir [Types de message](#) et [File d'attente de réponse et gestionnaire de files d'attente](#).

Pour plus d'informations sur les alias de file d'attente de réponse, voir [Alias de file d'attente de réponse et clusters](#).

## Utilisation de files d'attente alias

Vous pouvez définir une file d'attente alias pour faire référence indirectement à une autre file d'attente ou rubrique.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).



**Avertissement :** Les listes de distribution ne prennent pas en charge les files d'attente alias qui pointent vers des objets de rubrique. Si une file d'attente alias renvoie vers un objet de rubrique dans une liste de distribution, IBM MQ renvoie MQRC\_ALIAS\_BASE\_Q\_TYPE\_ERROR.

### Pourquoi et quand exécuter cette tâche

La file d'attente à laquelle une file d'attente alias fait référence peut être l'une des suivantes:

- Une file d'attente locale (voir «[Définition d'une file d'attente locale avec DEFINE QLOCAL](#)», à la page [147](#)).
- Définition locale d'une file d'attente éloignée (voir «[Utilisation des files d'attente éloignées](#)», à la page [157](#)).
- Rubrique.

Une file d'attente alias n'est pas une file d'attente réelle, mais une définition qui se résout en file d'attente réelle (ou cible) lors de l'exécution. La définition de file d'attente alias indique la file d'attente cible. Lorsqu'une application effectue un appel MQOPEN vers une file d'attente alias, le gestionnaire de files d'attente résout l'alias en nom de file d'attente cible.

Une file d'attente alias ne peut pas être résolue en une autre file d'attente alias définie localement. Toutefois, une file d'attente alias peut être convertie en files d'attente alias définies ailleurs dans des clusters dont le gestionnaire de files d'attente local est membre. Pour plus d'informations, voir [Résolution de nom](#).

Les files d'attente alias sont utiles pour:

- Attribution à différentes applications de différents niveaux de droits d'accès à la file d'attente cible.
- Permettre à différentes applications de travailler avec la même file d'attente de différentes manières. (Vous souhaitez peut-être affecter des priorités par défaut différentes ou des valeurs de persistance par défaut différentes.)
- Simplification de la maintenance, de la migration et de l'équilibrage de la charge de travail. (Vous souhaitez peut-être modifier le nom de la file d'attente cible sans avoir à modifier votre application, qui continue d'utiliser l'alias.)

Par exemple, supposons qu'une application ait été développée pour placer des messages dans une file d'attente appelée MY.ALIAS.QUEUE. Il indique le nom de cette file d'attente lorsqu'il émet une demande MQOPEN et, indirectement, s'il insère un message dans cette file d'attente. L'application ne sait pas que



la file d'attente est une file d'attente d'alias. Pour chaque appel MQI utilisant cet alias, le gestionnaire de files d'attente résout le nom de la file d'attente réelle, qui peut être une file d'attente locale ou une file d'attente éloignée définie sur ce gestionnaire de files d'attente.

En modifiant la valeur de l'attribut TARGET, vous pouvez rediriger les appels MQI vers une autre file d'attente, éventuellement sur un autre gestionnaire de files d'attente. Cela est utile pour la maintenance, la migration et l'équilibrage de charge.

## Procédure

- Définissez une file d'attente alias.

La commande MQSC suivante crée une file d'attente alias:

```
DEFINE QALIAS (MY.ALIAS.QUEUE) TARGET (YELLOW.QUEUE)
```

Cette commande redirige les appels MQI qui spécifient MY.ALIAS.QUEUE dans la file d'attente YELLOW.QUEUE. La commande ne crée pas la file d'attente cible ; les appels MQI échouent si la file d'attente est YELLOW.QUEUE n'existe pas lors de l'exécution.

Si vous modifiez la définition d'alias, vous pouvez rediriger les appels MQI vers une autre file d'attente. Exemple :

```
ALTER QALIAS (MY.ALIAS.QUEUE) TARGET (MAGENTA.QUEUE)
```

Cette commande redirige les appels MQI vers une autre file d'attente, MAGENTA.QUEUE.

Vous pouvez également utiliser des files d'attente alias pour qu'une seule file d'attente (la file d'attente cible) ait des attributs différents pour différentes applications. Pour ce faire, vous définissez deux alias, un pour chaque application. Supposons qu'il existe deux applications:

- L'application ALPHA peut placer des messages sur YELLOW.QUEUE, mais n'est pas autorisé à en extraire des messages.
- L'application bêta peut recevoir des messages de YELLOW.QUEUE, mais n'est pas autorisé à y placer des messages.

La commande MQSC suivante définit un alias qui est activé et désactivé pour l'application ALPHA:

```
DEFINE QALIAS (ALPHAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (ENABLED) +  
GET (DISABLED)
```

La commande suivante définit un alias qui est mis à l'état désactivé et activé pour l'application BETA:

```
DEFINE QALIAS (BETAS.ALIAS.QUEUE) +  
TARGET (YELLOW.QUEUE) +  
PUT (DISABLED) +  
GET (ENABLED)
```

ALPHA utilise le nom de file d'attente ALPHAS.ALIAS.QUEUE dans ses appels MQI ; BETA utilise le nom de file d'attente BETAS.ALIAS.QUEUE. Ils accèdent tous deux à la même file d'attente, mais de différentes manières.

Vous pouvez utiliser les attributs LIKE et REPLACE lorsque vous définissez des alias de file d'attente, de la même manière que vous utilisez ces attributs avec des files d'attente locales.

- Utilisez d'autres commandes avec des files d'attente alias.

Vous pouvez utiliser les commandes MQSC appropriées pour afficher ou modifier les attributs de file d'attente alias ou pour supprimer l'objet file d'attente alias. Exemple :

Utilisez la commande **DISPLAY QALIAS** pour afficher les attributs de la file d'attente alias:

```
DISPLAY QALIAS (ALPHAS.ALIAS.QUEUE)
```

Utilisez la commande **ALTER QALIAS** pour modifier le nom de la file d'attente de base, dans laquelle l'alias est résolu, où l'option `force` force la modification même si la file d'attente est ouverte:

```
ALTER QALIAS (ALPHAS.ALIAS.QUEUE) TARGET(ORANGE.LOCAL.QUEUE) FORCE
```

Utilisez la commande **DELETE QALIAS** pour supprimer cet alias de file d'attente:

```
DELETE QALIAS (ALPHAS.ALIAS.QUEUE)
```

Vous ne pouvez pas supprimer une file d'attente alias si la file d'attente est ouverte pour une application.

### Concepts associés

[Listes de diffusion](#)

### Référence associée

[ALTER QALIAS](#)

[DEFINE QALIAS](#)

[DELETE QALIAS](#)

## Utilisation des files d'attente modèles

Les files d'attente modèles fournissent une méthode pratique pour les applications afin de créer des files d'attente selon les besoins.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Pourquoi et quand exécuter cette tâche

Un gestionnaire de files d'attente crée une *file d'attente dynamique* s'il reçoit un appel MQI d'une application spécifiant un nom de file d'attente qui a été défini en tant que file d'attente modèle. Le nom de la nouvelle file d'attente dynamique est généré par le gestionnaire de files d'attente lors de la création de la file d'attente. Une *file d'attente modèle* est un modèle qui spécifie les attributs des files d'attente dynamiques créées à partir de celle-ci.

### Procédure

- Définissez une file d'attente modèle.

La commande **DEFINE QMODEL** MQSC permet de définir une file d'attente modèle avec un ensemble d'attributs de la même manière que vous définissez une file d'attente locale. Les files d'attente modèles et les files d'attente locales ont le même ensemble d'attributs, sauf que dans les files d'attente modèles, vous pouvez indiquer si les files d'attente dynamiques créées sont temporaires ou permanentes. (Les files d'attente permanentes sont gérées lors des redémarrages du gestionnaire de files d'attente, alors que les files d'attente temporaires ne le sont pas.) Exemple :

```
DEFINE QMODEL (GREEN.MODEL.QUEUE) +  
DESCR('Queue for messages from application X') +  
PUT (DISABLED) +  
GET (ENABLED) +  
NOTRIGGER +  
MSGDLVSQ (FIFO) +
```

```
MAXDEPTH (1000) +
MAXMSGL (2000) +
USAGE (NORMAL) +
DEFTYPE (PERMDYN)
```

Cette commande crée une définition de file d'attente modèle. A partir de l'attribut **DEFTYPE**, vous pouvez voir que les files d'attente réelles créées à partir de ce modèle sont des files d'attente dynamiques permanentes. Tous les attributs non spécifiés sont automatiquement copiés à partir de `SYSYTEM.DEFAULT.MODEL.QUEUE`.

Vous pouvez utiliser les attributs **LIKE** et **REPLACE** lorsque vous définissez des files d'attente modèles, de la même manière que vous les utilisez avec des files d'attente locales.

- Utilisez d'autres commandes avec des files d'attente modèles.

Vous pouvez utiliser les commandes MQSC appropriées pour afficher ou modifier les attributs d'une file d'attente modèle ou pour supprimer l'objet de file d'attente modèle. Exemple :

Utilisez la commande **DISPLAY QUEUE** pour afficher les attributs de la file d'attente modèle:

```
DISPLAY QUEUE (GREEN.MODEL.QUEUE)
```

Utilisez la commande **ALTER QMODEL** pour modifier le modèle afin d'activer les insertions dans n'importe quelle file d'attente dynamique créée à partir de ce modèle:

```
ALTER QMODEL (BLUE.MODEL.QUEUE) PUT(ENABLED)
```

Utilisez la commande **DELETE QMODEL** pour supprimer cette file d'attente modèle:

```
DELETE QMODEL (RED.MODEL.QUEUE)
```

### Référence associée

[ALTER QMODEL](#)

[DEFINIR QMODEL](#)

[SUPPRIMER QMODEL](#)

[DISPLAY QUEUE](#)

## Utilisation des files d'attente de rebut

Chaque gestionnaire de files d'attente possède généralement une file d'attente locale à utiliser comme file d'attente de rebut, de sorte que les messages qui ne peuvent pas être distribués à leur destination correcte puissent être stockés en vue d'une extraction ultérieure. Vous indiquez au gestionnaire de files d'attente la file d'attente de rebut et indiquez comment les messages trouvés dans une file d'attente de rebut doivent être traités. L'utilisation de files d'attente de rebut peut affecter la séquence dans laquelle les messages sont distribués. Vous pouvez donc choisir de ne pas les utiliser.

### Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Pourquoi et quand exécuter cette tâche

Exemple de file d'attente de rebut appelée `SYSTEM.DEAD.LETTER.QUEUE` est disponible avec le produit. Cette file d'attente est automatiquement créée lorsque vous créez le gestionnaire de files d'attente. Vous pouvez modifier cette définition si nécessaire et la renommer.

Une file d'attente de rebut n'a pas d'exigences particulières si ce n'est que:

- Il doit s'agir d'une file d'attente locale
- Son attribut MAXMSGL (longueur maximale de message) doit permettre à la file d'attente de recevoir les messages les plus volumineux que le gestionnaire de files d'attente doit traiter **plus** la taille de l'en-tête de message non livrés (MQDLH)

L'utilisation de files d'attente de rebut peut affecter la séquence dans laquelle les messages sont distribués. Vous pouvez donc choisir de ne pas les utiliser.

## Procédure

- Informez le gestionnaire de files d'attente de la file d'attente de rebut.

Pour ce faire, indiquez un nom de file d'attente de rebut dans la commande **crtmqm** (`crtmqm -u DEAD.LETTER.QUEUE`, par exemple) ou utilisez l'attribut **DEADQ** dans la commande **ALTER QMGR** pour en spécifier un ultérieurement. Vous devez définir la file d'attente de rebut avant de l'utiliser.

- Indiquez comment les messages trouvés dans une file d'attente de rebut doivent être traités.

Vous définissez l'attribut de canal USEDQ pour déterminer si la file d'attente de rebut est utilisée lorsque les messages ne peuvent pas être distribués. Cet attribut peut être configuré de sorte que certaines fonctions du gestionnaire de files d'attente utilisent la file d'attente de rebut alors que d'autres fonctions ne l'utilisent pas. Pour plus d'informations sur l'utilisation de l'attribut de canal USEDQ avec différentes commandes MQSC, voir [DEFINE CHANNEL](#), [DISPLAY CHANNEL](#), [ALTER CHANNEL](#) et [DISPLAY CLUSQMGR](#).

Le gestionnaire de files d'attente de rebut IBM MQ permet d'indiquer comment les messages trouvés dans une file d'attente de rebut doivent être traités ou supprimés. Voir [«Traitement des messages dans une file d'attente de rebut IBM MQ»](#), à la page 164.

### Concepts associés

[Files d'attente de rebut](#)

### Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

### Référence associée

[ALTER QMGR](#)

[crtmqm \(création du gestionnaire de files d'attente\)](#)

## Traitement des messages dans une file d'attente de rebut IBM MQ

Pour traiter des messages dans une file d'attente de rebut, utilisez le gestionnaire DLQ par défaut fourni par IBM MQ. Le gestionnaire compare les messages de la file d'attente des messages non livrés aux entrées d'une table de règles que vous définissez.

### Pourquoi et quand exécuter cette tâche

Les messages peuvent être placés sur un DLQ par des gestionnaires de files d'attente, des agents MCA et des applications. Tous les messages de la file d'attente de rebut doivent être précédés d'une structure *d'en-tête de non distribution*, MQDLH. Les messages insérés dans la file d'attente des messages non livrés par un gestionnaire de files d'attente ou un agent MCA ont toujours cet en-tête ; les applications qui placent des messages dans la file d'attente des messages non livrés doivent fournir cet en-tête. La zone *Reason* de la structure MQDLH contient un code anomalie indiquant la raison pour laquelle le message se trouve dans la file d'attente de rebut.

Tous les environnements IBM MQ ont besoin d'une routine pour traiter régulièrement les messages sur le DLQ. IBM MQ fournit une routine par défaut, appelée *gestionnaire de files d'attente de rebut* (gestionnaire DLQ), que vous appelez à l'aide de la commande **runmqdlq** MQSC.

Les instructions de traitement des messages sur le DLQ sont fournies au gestionnaire DLQ au moyen d'une *table de règles* écrite par l'utilisateur. En d'autres termes, le gestionnaire DLQ met en correspondance les messages de la DLQ avec les entrées de la table de règles ; lorsqu'un message

DLQ correspond à une entrée de la table de règles, le gestionnaire DLQ exécute l'action associée à cette entrée.

### Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

### Référence associée

[Files d'attente de rebut](#)

## Appel du gestionnaire de files d'attente de rebut

Appelez le gestionnaire de files d'attente de rebut (DLQ) à l'aide de la commande de contrôle de la file d'attente **runmqdlq**. Vous pouvez nommer le DLQ que vous souhaitez traiter et le gestionnaire de files d'attente que vous souhaitez utiliser de deux manières.

### Avant de commencer

Pour exécuter le gestionnaire DLQ, vous devez être autorisé à accéder à la fois à la file d'attente DLQ elle-même et aux files d'attente de messages vers lesquelles les messages de la file d'attente DLQ sont réacheminés. Pour que le gestionnaire DLQ puisse placer des messages dans des files d'attente avec les droits de l'ID utilisateur dans le contexte de message, vous devez également être autorisé à assumer l'identité des autres utilisateurs.

### Pourquoi et quand exécuter cette tâche

Les exemples suivants s'appliquent à la file d'attente des messages non livrés appelée ABC1.DEAD.LETTER.QUEUE, détenue par le gestionnaire de files d'attente ABC1.QUEUE.MANAGER.

Si vous ne spécifiez pas le DLQ ou le gestionnaire de files d'attente comme indiqué, le gestionnaire de files d'attente par défaut pour l'installation est utilisé avec le DLQ appartenant à ce gestionnaire de files d'attente.

La commande **runmqdlq** extrait ses entrées de `stdin`. Vous associez la table de règles à **runmqdlq** en redirigeant `stdin` à partir de la table de règles.

Pour plus d'informations sur la commande **runmqdlq**, voir [runmqdlq](#).

### Procédure

- Vous pouvez nommer le DLQ et le gestionnaire de files d'attente en tant que paramètres de la commande **runmqdlq**.

Par exemple, à partir de l'invite de commande:

```
runmqdlq ABC1.DEAD.LETTER.QUEUE ABC1.QUEUE.MANAGER <qrulc.rul
```

- Vous pouvez nommer le DLQ et le gestionnaire de files d'attente dans la table de règles.

Exemple :

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE) INPUTQM(ABC1.QUEUE.MANAGER)
```

### Concepts associés

[Files d'attente de rebut](#)

### Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

## Exemple de gestionnaire DLQ **amqsdlq**

En plus du gestionnaire de files d'attente de rebut appelé à l'aide de la commande **runmqdlq**, IBM MQ fournit la source d'un exemple de gestionnaire DLQ **amqsdlq** avec une fonction similaire à celle fournie par **runmqdlq**.

Vous pouvez personnaliser **amqsdlq** pour fournir un gestionnaire DLQ qui répond à vos besoins. Par exemple, vous pouvez décider que vous souhaitez un gestionnaire DLQ pouvant traiter des messages sans en-têtes de rebut. (Le gestionnaire DLQ par défaut et l'exemple, **amqsdlq**, traitent uniquement les messages du DLQ qui commencent par un en-tête de rebut, MQDLH. Les messages qui ne commencent pas par un MQDLH sont identifiés comme étant erronés et restent indéfiniment sur le DLQ.)

`MQ_INSTALLATION_PATH` représente le répertoire de haut niveau dans lequel IBM MQ est installé.

Dans IBM MQ for Windows, la source de **amqsdlq** est fournie dans le répertoire:

```
MQ_INSTALLATION_PATH\tools\c\samples\dlq
```

et la version compilée est fournie dans le répertoire:

```
MQ_INSTALLATION_PATH\tools\c\samples\bin
```

Sur les systèmes IBM MQ for UNIX et Linux, la source de **amqsdlq** est fournie dans le répertoire:

```
MQ_INSTALLATION_PATH/samp/dlq
```

et la version compilée est fournie dans le répertoire:

```
MQ_INSTALLATION_PATH/samp/bin
```

Une version générée de l'exemple de programme, nommé **amqsdlqc**, est incluse. Vous pouvez l'utiliser pour vous connecter à un gestionnaire de files d'attente éloignées en mode client. Pour utiliser **amqsdlqc**, vous devez définir l'une des variables d'environnement `MQSERVER`, `MQCHLLIB` ou `MQCHLTAB` pour identifier le mode de connexion au gestionnaire de files d'attente. Exemple :

```
export MQSERVER="SYSTEM.DEF.SVRCONN/TCP/myappliance.co.uk(1414) "
```

## Table des règles du gestionnaire DLQ

La table des règles du gestionnaire de files d'attente de rebut définit la façon dont le gestionnaire de files d'attente de rebut traite les messages qui arrivent sur la file d'attente de rebut.

Il existe deux types d'entrée dans une table de règles:

- La première entrée de la table, qui est facultative, contient des *données de contrôle*.
- Toutes les autres entrées de la table sont des *règles* que le gestionnaire DLQ doit suivre. Chaque règle se compose d'un *modèle* (un ensemble de caractéristiques de message) auquel un message est mis en correspondance, et d'une *action* à effectuer lorsqu'un message sur la file d'attente des messages non livrés correspond au modèle spécifié. Une table de règles doit contenir au moins une règle.

Chaque entrée de la table de règles comprend un ou plusieurs mots clés.

### Concepts associés

[Files d'attente de rebut](#)

### Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

## Données de contrôle DLQ

Vous pouvez inclure des mots clés dans une entrée de données de contrôle dans une table de règles de gestionnaire de files d'attente de rebut.

**Remarque :**

- La ligne verticale (|) sépare les alternatives, dont une seule peut être spécifiée.
- Tous les mots clés sont facultatifs.

### **INPUTQ ( QueueName|" (par défaut))**

Nom de la file d'attente des données à traiter:

1. Toute valeur INPUTQ que vous fournissez en tant que paramètre à la commande `runmqdlq` remplace toute valeur INPUTQ dans la table de règles.
2. Si vous ne spécifiez pas de valeur INPUTQ comme paramètre de la commande `runmqdlq`, mais que vous **indiquez** une valeur dans la table de règles, la valeur INPUTQ de la table de règles est utilisée.
3. Si aucun DLQ n'est spécifié ou que vous spécifiez INPUTQ (") dans la table de règles, le nom du DLQ appartenant au gestionnaire de files d'attente avec le nom fourni en tant que paramètre à la commande `runmqdlq` est utilisé.
4. Si vous ne spécifiez pas de valeur INPUTQ comme paramètre de la commande `runmqdlq` ou comme valeur dans la table de règles, le DLQ appartenant au gestionnaire de files d'attente nommé sur le mot clé INPUTQM dans la table de règles est utilisé.

### **INPUTQM ( QueueManagerNom|" (par défaut))**

Nom du gestionnaire de files d'attente propriétaire de la file d'attente des messages non livrés indiquée dans le mot clé INPUTQ:

1. Toute valeur INPUTQM que vous fournissez en tant que paramètre à la commande `runmqdlq` remplace toute valeur INPUTQM dans la table de règles.
2. Si vous ne spécifiez pas de valeur INPUTQM comme paramètre de la commande `runmqdlq`, la valeur INPUTQM de la table de règles est utilisée.
3. Si aucun gestionnaire de files d'attente n'est spécifié ou si vous spécifiez INPUTQM (") dans la table de règles, le gestionnaire de files d'attente par défaut pour l'installation est utilisé.

### **RETRYINT ( Intervalle| 60 (par défaut))**

Intervalle, en secondes, auquel le gestionnaire DLQ doit retraiter les messages sur le DLQ qui n'ont pas pu être traités lors de la première tentative et pour lesquels des tentatives répétées ont été demandées. Par défaut, l'intervalle entre les nouvelles tentatives est de 60 secondes.

### **WAIT ( YES (par défaut) |NO|nnn )**

Indique si le gestionnaire DLQ doit attendre que d'autres messages arrivent sur le DLQ lorsqu'il détecte qu'il ne peut pas traiter d'autres messages.

#### **YES**

Le gestionnaire DLQ attend indéfiniment.

#### **NO**

Le gestionnaire DLQ s'arrête lorsqu'il détecte que le DLQ est vide ou qu'il ne contient aucun message qu'il peut traiter.

#### **nnn**

Le gestionnaire DLQ attend *nnn* secondes pour que le nouveau travail arrive avant de s'arrêter, après avoir détecté que la file d'attente est vide ou qu'elle ne contient aucun message qu'il peut traiter.

Indiquez WAIT (YES) pour les DLQ occupés et WAIT (NO) ou WAIT (*nnn*) pour les DLQ dont le niveau d'activité est faible. Si le gestionnaire DLQ est autorisé à s'arrêter, appelez-le à nouveau à l'aide du déclenchement. Pour plus d'informations sur le déclenchement, voir [Démarriage des applications IBM MQ à l'aide de déclencheurs](#).

Une alternative à l'inclusion de données de contrôle dans la table de règles consiste à fournir les noms du DLQ et de son gestionnaire de files d'attente en tant que paramètres d'entrée à la commande `runmqdlq`. Si vous spécifiez une valeur à la fois dans la table de règles et en tant qu'entrée de la commande `runmqdlq`, la valeur spécifiée dans la commande `runmqdlq` est prioritaire.

Si vous incluez une entrée de données de contrôle dans la table de règles, il doit s'agir de la **première** entrée de la table.

Description des mots clés de correspondance de modèle (ceux auxquels correspondent les messages de la file d'attente des messages non livrés) et des mots clés d'action (ceux qui déterminent comment le gestionnaire DLQ doit traiter un message correspondant). Un exemple de règle est également fourni.

## Mots clés de correspondance de modèle

Les mots clés de correspondance de modèle, que vous utilisez pour spécifier les valeurs auxquelles les messages du DLQ sont mis en correspondance, sont les suivants. (Tous les mots clés de correspondance de modèle sont facultatifs):

### **APPLIDAT ( *DonnéesApplIdentity*|\* (valeur par défaut)**

Valeur *ApplIdentityData* spécifiée dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

### **APPLNAME ( *PutApplNom*|\* (par défaut))**

Nom de l'application qui a émis l'appel MQPUT ou MQPUT1 , comme indiqué dans la zone *PutApplNom* du descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

### **APPLTYPE ( *PutApplType*|\* (par défaut)**

Valeur *PutApplType* , spécifiée dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

### **DESTQ ( *QueueName*|\* (valeur par défaut))**

Nom de la file d'attente de messages à laquelle le message est destiné.

### **DESTQM ( *QueueManagerNom*|\* (par défaut))**

Nom du gestionnaire de files d'attente de la file d'attente de messages à laquelle le message est destiné.

### **FEEDBACK ( *Feedback*|\* (par défaut))**

Lorsque la valeur de *MsgType* est MQFB\_REPORT, *Feedback* décrit la nature du rapport.

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQFB\_COA pour identifier les messages sur le DLQ qui ont besoin d'une confirmation de leur arrivée dans leurs files d'attente de destination.

### **FORMAT ( *Format*|\* (par défaut))**

Nom utilisé par l'expéditeur du message pour décrire le format des données du message.

### **MSGTYPE ( *MsgType*|\* (valeur par défaut))**

Type du message sur la file d'attente des messages non livrés.

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQMT\_REQUEST pour identifier les messages sur le DLQ qui ont besoin de réponses.

### **PersIST ( *Persistence*|\* (par défaut))**

Valeur de persistance du message. (La persistance d'un message détermine s'il survit aux redémarrages du gestionnaire de files d'attente.)

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQPER\_PERSISTENT pour identifier les messages sur le DLQ qui sont persistants.

### **REASON ( *ReasonCode*|\* (par défaut))**

Code raison décrivant la raison pour laquelle le message a été inséré dans le DLQ.

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQRC\_Q\_FULL pour identifier les messages placés dans la file d'attente des messages non livrés car leurs files d'attente de destination étaient saturées.

### **REPLYQ ( *QueueName*|\* (par défaut))**

Nom de la file d'attente de réponses indiquée dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.



**REPLYQM ( *QueueManagerNom*|\* (par défaut))**

Nom du gestionnaire de files d'attente de la file d'attente de réponse, tel qu'indiqué dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

**USERID ( *UserIdentifieur*|\* (par défaut))**

ID de l'utilisateur à l'origine du message sur la file d'attente des messages non livrés, comme indiqué dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

**Les mots clés d'action**

Les mots clés d'action utilisés pour décrire la façon dont un message correspondant doit être traité sont les suivants:

**ACTION (DISCARD | IGNORE | RETRY | FWD)**

Action à effectuer pour tout message de la file d'attente des messages non livrés qui correspond au modèle défini dans cette règle.

**SUPPRIMER**

Supprimez le message du DLQ.

**IGNORE**

Laissez le message sur le DLQ.

**REESSAYER**

Si la première tentative d'insertion du message dans sa file d'attente de destination échoue, réessayez. Le mot clé RETRY définit le nombre de tentatives effectuées pour implémenter une action. Le mot clé RETRYINT des données de contrôle contrôle l'intervalle entre les tentatives.

**Transfert**

Réachemine le message vers la file d'attente indiquée dans le mot clé FWDQ.

Vous devez spécifier le mot clé ACTION.

**FWDQ ( *QueueName* | & DESTQ | & REPLYQ)**

Nom de la file d'attente de messages vers laquelle le message doit être réacheminé lorsque la commande ACTION (FWD) est demandée.

***QueueName***

Nom d'une file d'attente de messages. FWDQ (") n'est pas valide.

**& DESTQ**

Prenez le nom de la file d'attente dans la zone *DestQName* de la structure MQDLH.

**& REPLYQ**

Prenez le nom de la file d'attente dans la zone *ReplyToQ* du descripteur de message, MQMD.

Pour éviter les messages d'erreur lorsqu'une règle spécifiant FWDQ (& REPLYQ) correspond à un message dont la zone *ReplyToQ* est vide, spécifiez REPLYQ (? \*) dans le modèle de message.

**FWDQM ( *QueueManagerNom* | & DESTQM | & REPLYQM |" (par défaut))**

Gestionnaire de files d'attente de la file d'attente vers laquelle transférer un message.

***QueueManagerName***

Nom du gestionnaire de files d'attente de la file d'attente vers laquelle transférer un message lorsque ACTION (FWD) est demandée.

**& DESTQM**

Utilisez le nom du gestionnaire de files d'attente dans la zone *DestQMGrNom* de la structure MQDLH.

**& REPLYQM**

Utilisez le nom du gestionnaire de files d'attente dans la zone *ReplyToQMGr* du descripteur de message, MQMD.

..

FWDQM ("), qui est la valeur par défaut, identifie le gestionnaire de files d'attente local.

### **HEADER ( YES (par défaut) |NO)**

Indique si le MQDLH doit rester dans un message pour lequel ACTION (FWD) est demandée. Par défaut, le MQDLH reste sur le message. Le mot clé HEADER n'est pas valide pour les actions autres que FWD.

### **PUTAUT ( DEF (par défaut) | CTX )**

Droits avec lesquels les messages doivent être insérés par le gestionnaire DLQ:

#### **infrastructure d'évaluation de déploiement**

Placez les messages avec les droits du gestionnaire DLQ lui-même.

#### **CTX**

Placez les messages avec les droits de l'ID utilisateur dans le contexte de message. Si vous indiquez PUTAUT (CTX), vous devez être autorisé à assumer l'identité d'autres utilisateurs.

### **RETRY ( RetryCount|1 (par défaut))**

Nombre de fois, compris entre 1 et 999 999 999, où une action doit être exécutée (à l'intervalle indiqué dans le mot clé RETRYINT des données de contrôle). Le nombre de tentatives effectuées par le gestionnaire de file d'attente des messages non livrés pour implémenter une règle particulière est spécifique à l'instance en cours du gestionnaire de file d'attente des messages non livrés ; ce nombre n'est pas conservé lors des redémarrages. Si le gestionnaire DLQ est redémarré, le nombre de tentatives d'application d'une règle est réinitialisé à zéro.

## **Exemple de règle**

Voici un exemple de règle d'une table de règles de gestionnaire DLQ:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Cette règle demande au gestionnaire de file d'attente des messages non livrés d'effectuer trois tentatives de distribution à sa file d'attente de destination de tout message persistant inséré dans la file d'attente des messages non livrés car MQPUT et MQPUT1 ont été inhibés.

Tous les mots clés que vous pouvez utiliser sur une règle sont décrits dans le reste de cette section. Notez ce qui suit :

- La valeur par défaut d'un mot clé, le cas échéant, est soulignée. Pour la plupart des mots clés, la valeur par défaut est \* (astérisque), qui correspond à n'importe quelle valeur.
- La ligne verticale (|) sépare les alternatives, dont une seule peut être spécifiée.
- Tous les mots clés sauf ACTION sont facultatifs.

### Conventions de la table de règles DLQ

La syntaxe, la structure et le contenu de la table de règles du gestionnaire de files d'attente de rebut doivent respecter ces conventions.

La table de règles doit respecter les conventions suivantes:

- Une table de règles doit contenir au moins une règle.
- Les mots clés peuvent apparaître dans n'importe quel ordre.
- Un mot clé ne peut être inclus qu'une seule fois dans une règle.
- Les mots clés ne sont pas sensibles à la casse.
- Un mot clé et sa valeur de paramètre doivent être séparés des autres mots clés par au moins un espace ou une virgule.
- Il peut y avoir n'importe quel nombre de blancs au début ou à la fin d'une règle et entre les mots clés, la ponctuation et les valeurs.
- Chaque règle doit commencer sur une nouvelle ligne.

- Sur les systèmes Windows , la dernière règle de la table doit se terminer par un retour chariot / saut de ligne. Pour ce faire, vous devez appuyer sur la touche Entrée à la fin de la règle, de sorte que la dernière ligne de la table soit une ligne vide.
- Pour des raisons de portabilité, la longueur significative d'une ligne ne doit pas dépasser 72 caractères.
- Utilisez le signe plus (+) comme dernier caractère non blanc sur une ligne pour indiquer que la règle se poursuit à partir du premier caractère non blanc sur la ligne suivante. Utilisez le signe moins (-) comme dernier caractère non blanc sur une ligne pour indiquer que la règle se poursuit à partir du début de la ligne suivante. Les caractères de continuation peuvent apparaître dans les mots clés et les paramètres.

Exemple :

```
APPLNAME(' ABC+
D')
```

résultats dans 'ABCD', et

```
APPLNAME(' ABC-
D')
```

génère 'ABC D'.

- Les lignes de commentaire, qui commencent par un astérisque (\*), peuvent apparaître n'importe où dans la table de règles.
- Les lignes vides sont ignorées.
- Chaque entrée de la table des règles du gestionnaire DLQ comprend un ou plusieurs mots clés et leurs paramètres associés. Les paramètres doivent respecter les règles de syntaxe suivantes:
  - Chaque valeur de paramètre doit comporter au moins un caractère significatif. Les guillemets simples délimiteurs dans les valeurs placées entre guillemets ne sont pas considérés comme significatifs. Par exemple, les paramètres suivants sont valides:

FORMAT(' ABC')	3 caractères significatifs
FORMAT(ABC)	3 caractères significatifs
FORMAT(' A')	1 caractère significatif
FORMAT(A)	1 caractère significatif
FORMAT(' ')	1 caractère significatif

Ces paramètres ne sont pas valides car ils ne contiennent pas de caractères significatifs:

```
FORMAT(' ' )
FORMAT( )
FORMAT( )
FORMAT
```

- Les caractères génériques sont pris en charge. Vous pouvez utiliser le point d'interrogation (?) au lieu de n'importe quel caractère unique, à l'exception d'un blanc de fin ; vous pouvez utiliser l'astérisque (\*) au lieu de zéro ou plusieurs caractères adjacents. L'astérisque (\*) et le point d'interrogation (?) sont **toujours** interprétés comme des caractères génériques dans les valeurs de paramètre.
- Les caractères génériques ne peuvent pas être inclus dans les paramètres des mots clés suivants: ACTION, HEADER, RETRY, FWDQ, FWDQM et PUTAUT.
- Les blancs de fin dans les valeurs de paramètre et dans les zones correspondantes du message sur le DLQ ne sont pas importants lors de l'exécution de correspondances de caractères génériques. Toutefois, les blancs de début et les blancs imbriqués dans les chaînes placées entre apostrophes sont importants pour les correspondances de caractères génériques.

- Les paramètres numériques ne peuvent pas inclure le caractère générique point d'interrogation (?). Vous pouvez utiliser l'astérisque (\*) à la place d'un paramètre numérique entier, mais pas dans le cadre d'un paramètre numérique. Par exemple, les paramètres numériques suivants sont valides:

MSGTYPE(2)	Seuls les messages de réponse sont éligibles
MSGTYPE(*)	Tout type de message est éligible
MSGTYPE(' * ')	Tout type de message est éligible

Toutefois, MSGTYPE(' 2\* ') n'est pas valide car il inclut un astérisque (\*) dans un paramètre numérique.

- Les paramètres numériques doivent être compris entre 0 et 999 999 999. Si la valeur du paramètre est comprise dans cette plage, elle est acceptée, même si elle n'est pas valide dans la zone à laquelle le mot clé est associé. Vous pouvez utiliser des noms symboliques pour les paramètres numériques.
- Si une valeur de chaîne est plus courte que la zone du MQDLH ou du MQMD auquel le mot clé est associé, la valeur est remplie avec des blancs à la longueur de la zone. Si la valeur, à l'exclusion des astérisques, est plus longue que la zone, une erreur est diagnostiquée. Par exemple, voici toutes les valeurs de chaîne valides pour une zone de 8 caractères:

' ABCDEFGH '	8 caractères
' A*C*E*G*I '	5 caractères à l'exclusion des astérisques
' *A*C*E*G*I*K*M*O * '	8 caractères à l'exclusion des astérisques

- Placez entre guillemets simples les chaînes qui contiennent des blancs, des caractères minuscules ou des caractères spéciaux autres que le point (.), la barre oblique (?), le trait de soulignement ( \_ ) et le signe de pourcentage (%). Les caractères minuscules qui ne sont pas placés entre apostrophes sont convertis en majuscules. Si la chaîne inclut un guillemet, utilisez deux apostrophes pour indiquer le début et la fin du guillemet. Lorsque la longueur de la chaîne est calculée, chaque occurrence de guillemets est comptée comme un caractère unique.

#### Mode de traitement de la table de règles DLQ

Le gestionnaire de files d'attente de rebut recherche dans la table de règles une règle dans laquelle le modèle correspond à un message sur le DLQ.

La recherche commence par la première règle de la table et se poursuit séquentiellement dans la table. Lorsque le gestionnaire DLQ trouve une règle avec un modèle correspondant, il effectue l'action à partir de cette règle. Le gestionnaire DLQ incrémente le nombre de nouvelles tentatives pour une règle de 1 chaque fois qu'il applique cette règle. Si la première tentative échoue, le gestionnaire de la file d'attente des messages non livrés effectue une nouvelle tentative jusqu'à ce que le nombre de tentatives corresponde au nombre indiqué dans le mot clé RETRY. Si toutes les tentatives échouent, le gestionnaire DLQ recherche la règle de correspondance suivante dans la table.

Ce processus est répété pour les règles de correspondance suivantes jusqu'à ce qu'une action aboutisse. Lorsque chaque règle correspondante a été tentée le nombre de fois indiqué dans son mot clé RETRY et que toutes les tentatives ont échoué, ACTION (IGNORE) est pris en compte. ACTION (IGNORE) est également pris en compte si aucune règle de correspondance n'est trouvée.

#### Remarque :

1. Les modèles de règle de correspondance sont recherchés uniquement pour les messages de la file d'attente des messages non livrés qui commencent par un MQDLH. Les messages qui ne commencent pas par un MQDLH sont signalés périodiquement comme étant erronés et restent indéfiniment sur le DLQ.
2. Tous les mots clés de modèle peuvent être autorisés à être définis par défaut, de sorte qu'une règle ne peut être constituée que d'une action. Notez toutefois que les règles d'action uniquement sont appliquées à tous les messages de la file d'attente qui ont des MQDLH et qui n'ont pas déjà été traités conformément aux autres règles de la table.


3. La table de règles est validée lorsque le gestionnaire DLQ démarre et les erreurs sont signalées à ce moment-là. Vous pouvez apporter des modifications à la table de règles à tout moment, mais ces modifications n'entrent en vigueur que lorsque le gestionnaire DLQ redémarre.
4. Le gestionnaire DLQ ne modifie pas le contenu des messages, le MQDLH ou le descripteur de message. Le gestionnaire DLQ insère toujours des messages dans d'autres files d'attente avec l'option de message MQPMO\_PASS\_ALL\_CONTEXT.
5. Les erreurs de syntaxe consécutives dans la table de règles peuvent ne pas être reconnues car la table de règles est conçue pour éliminer la génération d'erreurs répétitives lors de la validation.
6. Le gestionnaire DLQ ouvre le DLQ avec l'option MQOO\_INPUT\_AS\_Q\_DEF.
7. Plusieurs instances du gestionnaire DLQ peuvent s'exécuter simultanément sur la même file d'attente, à l'aide de la même table de règles. Cependant, il est plus courant qu'il y ait une relation un à un entre un DLQ et un gestionnaire DLQ.

### Concepts associés

[Files d'attente de rebut](#)

### Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

 Exemple de table de règles de gestionnaire DLQ

Exemple de table de règles de file d'attente de rebut pour la commande **runmqdlq**, contenant une seule entrée de données de contrôle et plusieurs règles.

```
*****
*   An example rules table for the runmqdlq command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* runmqdlq, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to runmqdlq,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.
* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation are always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never do things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
```

```
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.
```

```
REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it should be able
* to cope with the message being lost, so we can afford to
* discard the message. PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We don't have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)

ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

## Concepts associés

[Files d'attente de rebut](#)

## Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

## Référence associée

[runmqdlq \(exécution du gestionnaire de files d'attente de rebut\)](#)

## Appel du gestionnaire de files d'attente de rebut sous IBM i

Sous IBM MQ for IBM i, vous appelez le gestionnaire DLQ en définissant la commande **STRMQMDLQ**.

## Avant de commencer

Vous devez être autorisé à accéder à la file d'attente des messages non livrés et aux files d'attente de messages auxquelles les messages de la file d'attente des messages non livrés sont transférés afin d'exécuter le gestionnaire de file d'attente des messages non livrés. Vous devez également être autorisé à utiliser l'identité d'autres utilisateurs pour que la file d'attente des messages soit placée dans des files d'attente avec les droits de l'ID utilisateur dans le contexte de message.

**Remarque :** Il est souvent préférable d'éviter de placer des messages sur un DLQ. Pour plus d'informations sur l'utilisation et l'évitement des DLQ, voir [«Utilisation des files d'attente de rebut»](#), à la page 163.

## Pourquoi et quand exécuter cette tâche

Une *file d'attente de messages non livrés* (DLQ), parfois appelée *file d'attente de messages non livrés*, est une file d'attente des messages qui ne peuvent pas être distribués à leurs files d'attente de destination. Chaque gestionnaire de files d'attente d'un réseau doit être associé à un DLQ.

Les gestionnaires de files d'attente, les agents MCA et les applications peuvent insérer des messages dans la file d'attente des messages non livrés. Tous les messages de la file d'attente de rebut doivent être précédés d'une structure d'*en-tête de non distribution*, MQDLH. Les messages insérés dans la file d'attente des messages non livrés par un gestionnaire de files d'attente ou par un agent de canal de

transmission de messages ont toujours un MQDLH. Pour les applications qui placent des messages sur le DLQ, vous devez fournir un MQDLH.

La zone *Reason* de la structure MQDLH contient un code anomalie indiquant la raison pour laquelle le message se trouve dans la file d'attente de rebut.

Dans tous les environnements IBM MQ, il doit y avoir une routine qui s'exécute régulièrement pour traiter les messages sur le DLQ. IBM MQ fournit une routine par défaut, appelée *gestionnaire de files d'attente de rebut* (gestionnaire DLQ), que vous appelez à l'aide de la commande **STRMQMDLQ**. Une *table de règles* écrite par l'utilisateur fournit des instructions au gestionnaire DLQ pour le traitement des messages sur le DLQ. En d'autres termes, le gestionnaire DLQ met en correspondance les messages de la file d'attente DLQ avec les entrées de la table de règles. Lorsqu'un message DLQ correspond à une entrée de la table des règles, le gestionnaire DLQ effectue l'action associée à cette entrée.

## Procédure

- Appeler le gestionnaire DLQ

Utilisez la commande **STRMQMDLQ** pour appeler le gestionnaire DLQ. Vous pouvez nommer la file d'attente des messages non livrés que vous souhaitez traiter et le gestionnaire de files d'attente que vous souhaitez utiliser de deux manières:

- En tant que paramètres de **STRMQMDLQ** à partir de l'invite de commande. Exemple :

```
STRMQMDLQ UDLMSGQ(ABC1.DEAD.LETTER.QUEUE) SRCMBR(QRULE) SRCFILE(library/QTXTSRC)
MQMNAME(MY.QUEUE.MANAGER)
```

- Dans la table de règles. Exemple :

```
INPUTQ(ABC1.DEAD.LETTER.QUEUE)
```

**Remarque :** La table de règles est un membre d'un fichier physique source qui peut prendre n'importe quel nom.

Les exemples s'appliquent à la file d'attente DLQ appelée ABC1.DEAD.LETTER.QUEUE, détenue par le gestionnaire de files d'attente par défaut.

Si vous ne spécifiez pas le DLQ ou le gestionnaire de files d'attente comme indiqué, le gestionnaire de files d'attente par défaut pour l'installation est utilisé avec le DLQ appartenant à ce gestionnaire de files d'attente.

## Concepts associés

[Files d'attente de rebut](#)

### Tâches associées

[Traitement des incidents liés aux messages non distribués](#)

### Table des règles du gestionnaire DLQ sous IBM i

La table des règles du gestionnaire de files d'attente de rebut définit comment le gestionnaire de files d'attente de rebut traite les messages qui arrivent sur la file d'attente de rebut IBM i.

La table des règles du gestionnaire DLQ définit comment le gestionnaire DLQ doit traiter les messages qui arrivent sur le DLQ. Il existe deux types d'entrée dans une table de règles:

- La première entrée de la table, qui est facultative, contient des *données de contrôle*.
- Toutes les autres entrées de la table sont des *règles* que le gestionnaire DLQ doit suivre. Chaque règle se compose d'un *modèle* (un ensemble de caractéristiques de message) auquel un message est mis en correspondance, et d'une *action* à effectuer lorsqu'un message sur la file d'attente des messages non livrés correspond au modèle spécifié. Une table de règles doit contenir au moins une règle.

Chaque entrée de la table de règles comprend un ou plusieurs mots clés.

## Données de contrôle

Cette section décrit les mots clés que vous pouvez inclure dans une entrée de données de contrôle dans une table de règles de gestionnaire DLQ. Notez ce qui suit :

- La valeur par défaut d'un mot clé, le cas échéant, est soulignée.
- La ligne verticale (|) sépare les alternatives. Vous ne pouvez spécifier qu'une seule de ces valeurs.
- Tous les mots clés sont facultatifs.

### **INPUTQ ( *QueueName*|" (par défaut))**

Nom de la file d'attente des données à traiter:

1. Toute valeur UDLMMSGQ (ou \*DFT) que vous indiquez comme paramètre de la commande **STRMQMDLQ** remplace toute valeur INPUTQ dans la table de règles.
2. Si vous indiquez une valeur UDLMMSGQ vide comme paramètre de la commande **STRMQMDLQ**, la valeur INPUTQ de la table de règles est utilisée.
3. Si vous indiquez une valeur UDLMMSGQ à blanc comme paramètre de la commande **STRMQMDLQ** et une valeur INPUTQ à blanc dans la table de règles, la file d'attente de rebut par défaut du système est utilisée.

### **INPUTQM ( *QueueManagerNom*|" (par défaut))**

Nom du gestionnaire de files d'attente propriétaire de la file d'attente des messages non livrés indiquée dans le mot clé INPUTQ.

Si vous ne spécifiez pas de gestionnaire de files d'attente ou si vous spécifiez INPUTQM (") dans la table de règles, le système utilise le gestionnaire de files d'attente par défaut pour l'installation.

### **RETRYINT ( *Intervalle*| 60 (par défaut))**

Intervalle, en secondes, auquel le gestionnaire DLQ doit tenter de retraiter les messages sur le DLQ qui n'ont pas pu être traités lors de la première tentative et pour lesquels des tentatives répétées ont été demandées. Par défaut, l'intervalle entre les nouvelles tentatives est de 60 secondes.

### **WAIT ( YES (par défaut) |NO|*nnn* )**

Indique si le gestionnaire DLQ doit attendre que d'autres messages arrivent sur le DLQ lorsqu'il détecte qu'il ne peut pas traiter d'autres messages.

#### **YES**

Provoque l'attente indéfinie du gestionnaire de la file d'attente des messages non livrés.

#### **NO**

Provoque l'arrêt du gestionnaire DLQ lorsqu'il détecte que le DLQ est vide ou qu'il ne contient aucun message qu'il peut traiter.


#### ***nnn***

Fait en sorte que le gestionnaire de la file d'attente des messages non livrés attende *nnn* secondes pour que le nouveau travail arrive avant de s'arrêter, après avoir détecté que la file d'attente est vide ou qu'elle ne contient aucun message qu'elle peut traiter.

Indiquez WAIT (YES) pour les DLQ occupés et WAIT (NO) ou WAIT ( *nnn* ) pour les DLQ dont le niveau d'activité est faible. Si le gestionnaire DLQ est autorisé à s'arrêter, appelez-le à nouveau à l'aide du déclenchement.

Vous pouvez indiquer le nom du DLQ comme paramètre d'entrée de la commande **STRMQMDLQ**, au lieu d'inclure des données de contrôle dans la table de règles. Si une valeur est spécifiée à la fois dans la table de règles et en entrée de la commande **STRMQMDLQ**, la valeur spécifiée dans la commande **STRMQMDLQ** est prioritaire.

**Remarque :** Si une entrée de données de contrôle est incluse dans la table de règles, elle doit être la première entrée de la table.

 Règles DLQ (modèles et actions) sur IBM i

Description des modèles et des actions pour chacune des règles de file d'attente de rebut IBM i.



Voici un exemple de règle d'une table de règles de gestionnaire DLQ:

```
PERSIST(MQPER_PERSISTENT) REASON (MQRC_PUT_INHIBITED) +  
ACTION (RETRY) RETRY (3)
```

Cette règle demande au gestionnaire DLQ d'effectuer 3 tentatives de distribution à sa file d'attente de destination de tout message persistant qui a été inséré dans la file d'attente DLQ car MQPUT et MQPUT1 ont été inhibés.

Cette section décrit les mots clés que vous pouvez inclure dans une règle. Notez ce qui suit :

- La valeur par défaut d'un mot clé, le cas échéant, est soulignée. Pour la plupart des mots clés, la valeur par défaut est \* (astérisque), qui correspond à n'importe quelle valeur.
- La ligne verticale (|) sépare les alternatives. Vous ne pouvez spécifier qu'une seule de ces valeurs.
- Tous les mots clés sauf ACTION sont facultatifs.

Cette section commence par une description des mots clés de correspondance de modèle (ceux avec lesquels les messages du DLQ sont mis en correspondance). Il décrit ensuite les mots clés d'action (ceux qui déterminent comment le gestionnaire DLQ doit traiter un message correspondant).

#### **IBM i** Mots clés de correspondance de modèle DLQ sous IBM i

Les mots clés de correspondance de modèle sont décrits dans un exemple. Utilisez ces mots clés pour spécifier les valeurs par rapport auxquelles les messages de la file d'attente de rebut IBM i sont mis en correspondance. Tous les mots clés de correspondance de modèle sont facultatifs.

#### **APPLIDAT ( *DonnéesApplIdentity* | \* (valeur par défaut)**

Valeur *ApplIdentityData* du message sur la file d'attente des messages non livrés, indiquée dans le descripteur de message, MQMD.

#### **APPLNAME ( *PutApplNom* | \* (par défaut)**

Nom de l'application qui a émis l'appel MQPUT ou MQPUT1 , comme indiqué dans la zone *PutApplName* du descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

#### **APPLTYPE ( *PutApplType* | \* (par défaut)**

Valeur *PutApplType* spécifiée dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

#### **DESTQ ( *QueueName* | \* (valeur par défaut)**

Nom de la file d'attente de messages à laquelle le message est destiné.

#### **DESTQM ( *QueueManagerNom* | \* (par défaut)**

Nom du gestionnaire de files d'attente de la file d'attente de messages à laquelle le message est destiné.

#### **FEEDBACK ( *Feedback* | \* (par défaut)**

Lorsque la valeur *MsgType* est MQMT\_REPORT, *Feedback* décrit la nature du rapport.

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQFB\_COA pour identifier les messages de la file d'attente des messages non livrés qui nécessitent une confirmation de leur arrivée dans leurs files d'attente de destination.

#### **FORMAT ( *Format* | \* (par défaut)**

Nom utilisé par l'expéditeur du message pour décrire le format des données du message.

#### **MSGTYPE ( *MsgType* | \* (valeur par défaut)**

Type du message sur la file d'attente des messages non livrés.

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQMT\_REQUEST pour identifier les messages sur le DLQ qui nécessitent des réponses.

**PersIST ( *Persistence* | \* (par défaut))**

Valeur de persistance du message. (La persistance d'un message détermine s'il survit aux redémarrages du gestionnaire de files d'attente.)

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQPER\_PERSISTENT pour identifier les messages sur le DLQ qui sont persistants.

**REASON ( *ReasonCode* | \* (par défaut))**

Code raison décrivant la raison pour laquelle le message a été inséré dans le DLQ.

Vous pouvez utiliser des noms symboliques. Par exemple, vous pouvez utiliser le nom symbolique MQRC\_Q\_FULL pour identifier les messages placés dans la file d'attente des messages non livrés car leurs files d'attente de destination étaient saturées.

**REPLYQ ( *QueueName* | \* (par défaut))**

Nom de la file d'attente de réponse indiqué dans le descripteur de message, MQMD, du message sur la file d'attente des messages non livrés.

**REPLYQM ( *QueueManagerNom* | \* (par défaut))**

Nom du gestionnaire de files d'attente de la file d'attente de réponses spécifiée dans le mot clé REPLYQ.

**USERID ( *UserIdentifier* | \* (par défaut))**

ID de l'utilisateur à l'origine du message sur la file d'attente des messages non livrés, comme indiqué dans le descripteur de message, MQMD.

 **Mots clés d'action DLQ sous IBM i**

Utilisez ces mots clés d'action de file d'attente de rebut pour déterminer comment un message correspondant dans la file d'attente de rebut IBM i est traité.

**ACTION (DISCARD | IGNORE | RETRY | FWD)**

Action effectuée pour tout message de la file d'attente des messages non livrés qui correspond au modèle défini dans cette règle.

**SUPPRIMER**

Entraîne la suppression du message de la file d'attente des messages non livrés.

**IGNORE**

Permet de conserver le message dans la file d'attente des messages non livrés.

**REESSAYER**

Indique au gestionnaire de la file d'attente des messages non livrés de tenter à nouveau d'insérer le message dans sa file d'attente de destination.

**Transfert**

Permet au gestionnaire DLQ de transmettre le message à la file d'attente indiquée dans le mot clé FWDQ.

Vous devez spécifier le mot clé ACTION. Le nombre de tentatives d'implémentation d'une action est régi par le mot clé RETRY. Le mot clé RETRYINT des données de contrôle contrôle l'intervalle entre les tentatives.

**FWDQ ( *QueueName* | & DESTQ | & REPLYQ)**

Nom de la file d'attente de messages à laquelle le message est réacheminé lorsque vous sélectionnez le mot clé ACTION.

***QueueName***

Nom d'une file d'attente de messages. FWDQ (") n'est pas valide.

**& DESTQ**

Prenez le nom de la file d'attente dans la zone *DestQName* de la structure MQDLH.

**& REPLYQ**

Prenez le nom de la file d'attente dans la zone *ReplyToQ* du descripteur de message, MQMD.

Vous pouvez spécifier REPLYQ (? \*) dans le modèle de message pour éviter les messages d'erreur, lorsqu'une règle spécifiant FWDQ (& REPLYQ) correspond à un message dont la zone *ReplyToQ* est vide.

### **FWDQM ( *QueueManagerNom* | & DESTQM | & REPLYQM |" (par défaut))**

Gestionnaire de files d'attente de la file d'attente vers laquelle un message est réacheminé.

#### ***QueueManagerName***

Nom du gestionnaire de files d'attente de la file d'attente vers laquelle le message est réacheminé lorsque vous sélectionnez le mot clé ACTION (FWD).

#### **& DESTQM**

Utilisez le nom du gestionnaire de files d'attente dans la zone *DestQMGrNom* de la structure MQDLH.

#### **& REPLYQM**

Utilisez le nom du gestionnaire de files d'attente dans la zone *ReplyToQMGr* du descripteur de message, MQMD.

..

FWDQM ("), qui est la valeur par défaut, identifie le gestionnaire de files d'attente local.

### **HEADER ( YES (par défaut) |NO)**

Indique si le MQDLH doit rester dans un message pour lequel ACTION (FWD) est demandée. Par défaut, le MQDLH reste sur le message. Le mot clé HEADER n'est pas valide pour les actions autres que FWD.

### **PUTAUT (DEF (par défaut) | CTX)**

Droits avec lesquels les messages doivent être insérés par le gestionnaire DLQ:

#### **infrastructure d'évaluation de déploiement**

Insère des messages avec les droits du gestionnaire DLQ lui-même.

#### **CTX**

Permet d'insérer les messages avec les droits de l'ID utilisateur dans le contexte de message. Vous devez être autorisé à assumer l'identité d'autres utilisateurs, si vous indiquez PUTAUT (CTX).

### **RETRY ( *RetryCount*|1 (par défaut))**

Nombre de fois, compris entre 1 et 999 999 999, où une action doit être tentée (à l'intervalle indiqué dans le mot clé RETRYINT des données de contrôle).

**Remarque :** Le nombre de tentatives effectuées par le gestionnaire de file d'attente des messages non livrés pour implémenter une règle particulière est spécifique à l'instance en cours du gestionnaire de file d'attente des messages non livrés ; ce nombre n'est pas conservé lors des redémarrages. Si vous redémarrez le gestionnaire DLQ, le nombre de tentatives d'application d'une règle est réinitialisé à zéro.

### **IBM i** *Conventions de table de règles DLQ sous IBM i*

La table de règles de file d'attente de rebut IBM i doit respecter des conventions spécifiques concernant sa syntaxe, sa structure et son contenu.

- Une table de règles doit contenir au moins une règle.
- Les mots clés peuvent apparaître dans n'importe quel ordre.
- Un mot clé ne peut être inclus qu'une seule fois dans une règle.
- Les mots clés ne sont pas sensibles à la casse.
- Un mot clé et sa valeur de paramètre doivent être séparés des autres mots clés par au moins un espace ou une virgule.
- N'importe quel nombre de blancs peut apparaître au début ou à la fin d'une règle et entre les mots clés, la ponctuation et les valeurs.
- Chaque règle doit commencer sur une nouvelle ligne.
- Pour la portabilité, la longueur significative d'une ligne ne doit pas dépasser 72 caractères.

- Utilisez le signe plus (+) comme dernier caractère non blanc sur une ligne pour indiquer que la règle se poursuit à partir du premier caractère non blanc sur la ligne suivante. Utilisez le signe moins (-) comme dernier caractère non blanc sur une ligne pour indiquer que la règle se poursuit à partir du début de la ligne suivante. Les caractères de continuation peuvent apparaître dans les mots clés et les paramètres.

Exemple :

```
APPLNAME (' ABC+
D')
```

gène'ABCD'.

```
APPLNAME (' ABC-
D')
```

gène'ABC D'.

- Les lignes de commentaire, qui commencent par un astérisque (\*), peuvent apparaître n'importe où dans la table de règles.
- Les lignes vides sont ignorées.
- Chaque entrée de la table des règles du gestionnaire DLQ comprend un ou plusieurs mots clés et leurs paramètres associés. Les paramètres doivent respecter les règles de syntaxe suivantes:
  - Chaque valeur de paramètre doit comporter au moins un caractère significatif. Les guillemets de délimitation dans les valeurs placées entre guillemets ne sont pas considérés comme significatifs. Par exemple, les paramètres suivants sont valides:

FORMAT (' ABC ')	3 caractères significatifs
FORMAT (ABC)	3 caractères significatifs
FORMAT (' A ')	1 caractère significatif
FORMAT (A)	1 caractère significatif
FORMAT ( ' ')	1 caractère significatif

Ces paramètres ne sont pas valides car ils ne contiennent pas de caractères significatifs:

```
FORMAT ( ' ' )
FORMAT ( )
FORMAT ( )
FORMAT
```

- Les caractères génériques sont pris en charge. Vous pouvez utiliser le point d'interrogation (?) à la place d'un caractère unique, à l'exception d'un blanc de fin. Vous pouvez utiliser l'astérisque (\*) à la place de zéro ou plusieurs caractères adjacents. L'astérisque (\*) et le point d'interrogation (?) sont **toujours** interprétés comme des caractères génériques dans les valeurs de paramètre.
- Vous ne pouvez pas inclure de caractères génériques dans les paramètres des mots clés suivants: ACTION, HEADER, RETRY, FWDQ, FWDQM et PUTAUT.
- Les blancs de fin dans les valeurs de paramètre et dans les zones correspondantes du message sur le DLQ ne sont pas importants lors de l'exécution de correspondances de caractères génériques. Toutefois, les blancs de début et les blancs imbriqués dans les chaînes entre guillemets sont importants pour les correspondances de caractères génériques.
- Les paramètres numériques ne peuvent pas inclure le caractère générique point d'interrogation (?). Vous pouvez inclure l'astérisque (\*) à la place d'un paramètre numérique entier, mais l'astérisque ne peut pas être inclus dans un paramètre numérique. Par exemple, les paramètres numériques suivants sont valides:

MSGTYPE(2)	Seuls les messages de réponse sont éligibles
MSGTYPE(*)	Tout type de message est éligible
MSGTYPE(' *')	Tout type de message est éligible

Toutefois, MSGTYPE('2\*') n'est pas valide car il inclut un astérisque (\*) dans un paramètre numérique.

- Les paramètres numériques doivent être compris entre 0 et 999 999 999. Si la valeur du paramètre est comprise dans cette plage, elle est acceptée, même si elle n'est pas valide dans la zone à laquelle le mot clé est associé. Vous pouvez utiliser des noms symboliques pour les paramètres numériques.
- Si une valeur de chaîne est plus courte que la zone du MQDLH ou du MQMD auquel le mot clé est associé, la valeur est remplie avec des blancs à la longueur de la zone. Si la valeur, à l'exclusion des astérisques, est plus longue que la zone, une erreur est diagnostiquée. Par exemple, voici toutes les valeurs de chaîne valides pour une zone de 8 caractères:

'ABCDEFGH'	8 caractères
'A*C*E*G*I'	5 caractères à l'exclusion des astérisques
'*A*C*E*G*I*K*M*O*'	8 caractères à l'exclusion des astérisques

- Les chaînes contenant des blancs, des caractères minuscules ou des caractères spéciaux autres que le point (.), la barre oblique (?), le trait de soulignement (\_) et le signe de pourcentage (%) doivent être placés entre guillemets simples. Les caractères minuscules qui ne sont pas placés entre guillemets sont convertis en majuscules. Si la chaîne comprend un guillemet, deux apostrophes doivent être utilisées pour indiquer à la fois le début et la fin du guillemet. Lorsque la longueur de la chaîne est calculée, chaque occurrence de guillemets est comptée comme un caractère unique.

#### *Traitement de la table de règles DLQ sur IBM i*

Le gestionnaire de files d'attente de rebut recherche dans la table de règles une règle dont le modèle correspond à un message de la file d'attente de rebut IBM i.

La recherche commence par la première règle de la table et se poursuit séquentiellement dans la table. Lorsqu'une règle avec un modèle de correspondance est trouvée, la table de règles tente l'action à partir de cette règle. Le gestionnaire DLQ incrémente le nombre de nouvelles tentatives pour une règle de 1 chaque fois qu'il tente d'appliquer cette règle. Si la première tentative échoue, la tentative est répétée jusqu'à ce que le nombre de tentatives effectuées corresponde au nombre indiqué dans le mot clé RETRY. Si toutes les tentatives échouent, le gestionnaire DLQ recherche la règle de correspondance suivante dans la table.

Ce processus est répété pour les règles de correspondance suivantes jusqu'à ce qu'une action aboutisse. Lorsque chaque règle correspondante a été tentée le nombre de fois indiqué dans son mot clé RETRY et que toutes les tentatives ont échoué, ACTION (IGNORE) est pris en compte. ACTION (IGNORE) est également pris en compte si aucune règle de correspondance n'est trouvée.

#### **Remarque :**

1. Les modèles de règle de correspondance sont recherchés uniquement pour les messages de la file d'attente des messages non livrés qui commencent par un MQDLH. Les messages qui ne commencent pas par un MQDLH sont signalés périodiquement comme étant erronés et restent indéfiniment sur le DLQ.
2. Tous les mots clés de modèle peuvent être définis par défaut, de sorte qu'une règle ne peut être constituée que d'une action. Notez toutefois que les règles d'action uniquement sont appliquées à tous les messages de la file d'attente qui ont des MQDLH et qui n'ont pas déjà été traités conformément aux autres règles de la table.
3. La table de règles est validée lorsque le gestionnaire DLQ démarre et les erreurs sont signalées à ce moment-là. (Les messages d'erreur émis par le gestionnaire DLQ sont décrits dans [Messages et codes anomalie](#).) Vous pouvez apporter des modifications à la table de règles à tout moment, mais ces modifications n'entrent pas en vigueur tant que le gestionnaire DLQ n'est pas redémarré.

4. Le gestionnaire DLQ ne modifie pas le contenu des messages, du MQDLH ou du descripteur de message. Le gestionnaire DLQ insère toujours des messages dans d'autres files d'attente avec l'option de message **MQPMO\_PASS\_ALL\_CONTEXT**.
5. Les erreurs de syntaxe consécutives dans la table de règles peuvent ne pas être reconnues, car la validation de la table de règles élimine la génération d'erreurs répétitives.
6. Le gestionnaire DLQ ouvre le DLQ avec l'option **MQOO\_INPUT\_AS\_Q\_DEF**.
7. Plusieurs instances du gestionnaire DLQ peuvent s'exécuter simultanément sur la même file d'attente, à l'aide de la même table de règles. Cependant, il est plus courant qu'il y ait une relation un à un entre un DLQ et un gestionnaire DLQ.

**IBM i**

*Exemple de table de règles de gestionnaire DLQ sous IBM i*

Exemple de code pour une table de règles de gestionnaire de files d'attente de rebut sous IBM i. Cet exemple de table de règles contient une entrée de données de contrôle unique et plusieurs règles.

```
*****
*   An example rules table for the STRMQMDLQ command   *
*****
* Control data entry
* -----
* If no queue manager name is supplied as an explicit parameter to
* STRMQMDLQ, use the default queue manager for the machine.
* If no queue name is supplied as an explicit parameter to STRMQMDLQ,
* use the DLQ defined for the local queue manager.
*
inputqm(' ') inputq(' ')

* Rules
* -----
* We include rules with ACTION (RETRY) first to try to
* deliver the message to the intended destination.

* If a message is placed on the DLQ because its destination
* queue is full, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_Q_FULL) ACTION(RETRY) RETRY(5)

* If a message is placed on the DLQ because of a put inhibited
* condition, attempt to forward the message to its
* destination queue. Make 5 attempts at approximately
* 60-second intervals (the default value for RETRYINT).

REASON(MQRC_PUT_INHIBITED) ACTION(RETRY) RETRY(5)

* The AAAA corporation is always sending messages with incorrect
* addresses. When we find a request from the AAAA corporation,
* we return it to the DLQ (DEADQ) of the reply-to queue manager
* (&REPLYQM).
* The AAAA DLQ handler attempts to redirect the message.

MSGTYPE(MQMT_REQUEST) REPLYQM(AAAA.*) +
ACTION(FWD) FWDQ(DEADQ) FWDQM(&REPLYQM)

* The BBBB corporation never does things by half measures. If
* the queue manager BBBB.1 is unavailable, try to
* send the message to BBBB.2

DESTQM(bbbb.1) +
action(fwd) fwdq(&DESTQ) fwdqm(bbbb.2) header(no)

* The CCCC corporation considers itself very security
* conscious, and believes that none of its messages
* will ever end up on one of our DLQs.
* Whenever we see a message from a CCCC queue manager on our
* DLQ, we send it to a special destination in the CCCC organization
* where the problem is investigated.

REPLYQM(CCCC.*) +
ACTION(FWD) FWDQ(ALARM) FWDQM(CCCC.SYSTEM)
```

```
* Messages that are not persistent run the risk of being
* lost when a queue manager terminates. If an application
* is sending nonpersistent messages, it must be able
* to cope with the message being lost, so we can afford to
* discard the message.
```

```
PERSIST(MQPER_NOT_PERSISTENT) ACTION(DISCARD)
```

```
* For performance and efficiency reasons, we like to keep
* the number of messages on the DLQ small.
* If we receive a message that has not been processed by
* an earlier rule in the table, we assume that it
* requires manual intervention to resolve the problem.
* Some problems are best solved at the node where the
* problem was detected, and others are best solved where
* the message originated. We do not have the message origin,
* but we can use the REPLYQM to identify a node that has
* some interest in this message.
* Attempt to put the message onto a manual intervention
* queue at the appropriate node. If this fails,
* put the message on the manual intervention queue at
* this node.
```

```
REPLYQM('?*') +
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION) FWDQM(&REPLYQM)
```

```
ACTION(FWD) FWDQ(DEADQ.MANUAL.INTERVENTION)
```

## Vérification du traitement de tous les messages DLQ

Le gestionnaire de file d'attente de rebut conserve un enregistrement de tous les messages de la file d'attente des messages non livrés qui ont été vus mais qui n'ont pas été supprimés. Assurez-vous que le DLQ contient le moins de messages possible.

## Pourquoi et quand exécuter cette tâche

Si vous utilisez le gestionnaire DLQ comme filtre pour extraire un petit sous-ensemble des messages du DLQ, le gestionnaire DLQ conserve toujours un enregistrement de ces messages sur le DLQ qu'il n'a pas traité. En outre, le gestionnaire DLQ ne peut pas garantir que les nouveaux messages arrivant sur le DLQ sont vus, même si le DLQ est défini comme premier entré, premier sorti (FIFO). Si la file d'attente n'est pas vide, le DLQ est périodiquement réanalysé pour vérifier tous les messages.

Pour ces raisons, vous devez vous assurer que le DLQ contient le moins de messages possible. Si des messages qui ne peuvent pas être supprimés ou réacheminés vers d'autres files d'attente (pour quelque raison que ce soit) sont autorisés à s'accumuler dans la file d'attente, la charge de travail du gestionnaire DLQ augmente et la file d'attente DLQ elle-même risque de se remplir.

Pour permettre au gestionnaire DLQ de vider le DLQ, procédez comme suit:

## Procédure

- Pour les messages que vous ignorez autrement, utilisez une action qui déplace les messages vers une autre file d'attente.

Essayez de ne pas utiliser la commande **ACTION (IGNORE)**, qui laisse les messages sur la file d'attente des messages non livrés, et n'oubliez pas que **ACTION (IGNORE)** est supposé être utilisé pour les messages qui ne sont pas explicitement adressés par d'autres règles de la table. A la place, utilisez une action qui déplace les messages vers une autre file d'attente. Exemple :

```
ACTION (FWD) FWDQ (IGNORED.DEAD.QUEUE) HEADER (YES)
```

- Faites de la règle finale de la table un "catch-all" pour traiter les messages qui n'ont pas été traités par les règles précédentes de la table.

Par exemple, la règle finale de la table peut ressembler à ceci:

```
ACTION (FWD) FWDQ (REALLY.DEAD.QUEUE) HEADER (YES)
```

Les messages qui tombent dans la règle finale de la table sont ainsi réacheminé vers la file d'attente REALLY . DEAD . QUEUE, où ils peuvent être traités manuellement. Si vous n'avez pas de règle de ce type, les messages risquent de rester indéfiniment dans la file d'attente des messages non livrés.

## Utilisation des rubriques d'administration

Utilisez les commandes MQSC pour gérer les rubriques d'administration.

Pour plus d'informations sur ces commandes, voir [Commandes MQSC](#) .

### Concepts associés

[Objets de rubrique d'administration](#)

### Tâches associées

«Définition d'une rubrique d'administration», à la page [184](#)

Utilisez la commande MQSC **DEFINE TOPIC** pour créer une rubrique d'administration. Lorsque vous définissez une rubrique d'administration, vous pouvez éventuellement définir chaque attribut de rubrique.

«Affichage des attributs d'objet de rubrique d'administration», à la page [185](#)

La commande MQSC **DISPLAY TOPIC** permet d'afficher un objet de rubrique d'administration.

«Modification des attributs de rubrique d'administration», à la page [186](#)

Vous pouvez modifier les attributs de rubrique de deux manières, à l'aide de la commande **ALTER TOPIC** ou de la commande **DEFINE TOPIC** avec l'attribut **REPLACE** .

«Copie d'une définition de rubrique d'administration», à la page [186](#)

Vous pouvez copier une définition de rubrique à l'aide de l'attribut LIKE de la commande **DEFINE** .

«Suppression d'une définition de rubrique d'administration», à la page [187](#)

Vous pouvez utiliser la commande MQSC **DELETE TOPIC** pour supprimer une rubrique d'administration.

## Définition d'une rubrique d'administration

Utilisez la commande MQSC **DEFINE TOPIC** pour créer une rubrique d'administration. Lorsque vous définissez une rubrique d'administration, vous pouvez éventuellement définir chaque attribut de rubrique.

### Avant de commencer

**Remarque :** L'exemple de cette tâche nécessite que vous exécutiez des commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Pourquoi et quand exécuter cette tâche

Tout attribut de la rubrique qui n'est pas explicitement défini est hérité de la rubrique d'administration par défaut, SYSTEM.DEFAULT.TOPIC, qui a été créé lors de l'installation du système.

### Exemple

Par exemple, la commande **DEFINE TOPIC** qui suit, définit une rubrique appelée ORANGE . TOPIC avec les caractéristiques suivantes:

- Se résout en la chaîne de rubrique ORANGE. Pour plus d'informations sur la façon dont les chaînes de rubrique peuvent être utilisées, voir [Combinaison de chaînes de rubrique](#).
- Tout attribut défini sur ASPARENT utilise l'attribut tel que défini par la rubrique parent de cette rubrique. Cette action est répétée dans l'arborescence de rubriques jusqu'à la rubrique racine, SYSTEM.BASE.TOPIC a été trouvé. Pour plus d'informations, voir [Arborescences de rubriques](#).

```
DEFINE TOPIC (ORANGE.TOPIC) +  
TOPICSTR (ORANGE) +  
DEFPRTY(ASPARENT) +  
NPMSGDLV(ASPARENT)
```



## Remarque :

- A l'exception de la valeur de la chaîne de rubrique, toutes les valeurs d'attribut affichées sont les valeurs par défaut. Ils ne sont présentés ici qu'à titre d'illustration. Vous pouvez les omettre si vous êtes sûr que les valeurs par défaut correspondent à celles que vous souhaitez ou qu'elles n'ont pas été modifiées. Voir aussi «Affichage des attributs d'objet de rubrique d'administration», à la page 185.
- Si vous disposez déjà d'une rubrique d'administration sur le même gestionnaire de files d'attente avec le nom ORANGE.TOPIC, cette commande échoue. Utilisez l'attribut REPLACE si vous souhaitez remplacer la définition existante d'une rubrique, mais voir aussi «Modification des attributs de rubrique d'administration», à la page 186.

## Référence associée

[DEFINE TOPIC](#)

## Affichage des attributs d'objet de rubrique d'administration

La commande MQSC **DISPLAY TOPIC** permet d'afficher un objet de rubrique d'administration.

## Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

## Exemple

Cette commande affiche toutes les rubriques:

```
DISPLAY TOPIC(ORANGE.TOPIC)
```

Vous pouvez afficher les attributs de manière sélective en les spécifiant individuellement à l'aide de la commande **DISPLAY TOPIC** . Exemple :

```
DISPLAY TOPIC(ORANGE.TOPIC) +  
TOPICSTR +  
DEFPRTY +  
NPMSGDLV
```

Cette commande affiche les trois attributs spécifiés:

```
AMQ8633: Display topic details.  
TOPIC(ORANGE.TOPIC) TYPE(LOCAL)  
TOPICSTR(ORANGE) DEFPRTY(ASPARENT)  
NPMSGDLV(ASPARENT)
```

Pour afficher les valeurs ASPARENT de la rubrique telles qu'elles sont utilisées lors de l'exécution, utilisez la commande **DISPLAY TPSTATUS** . Par exemple, utilisez :

```
DISPLAY TPSTATUS(ORANGE) DEFPRTY NPMSGDLV
```

La commande affiche les détails suivants:

```
AMQ8754: Display topic status details.  
TOPICSTR(ORANGE) DEFPRTY(0)  
NPMSGDLV(ALLAVAIL)
```

Lorsque vous définissez une rubrique d'administration, elle prend tous les attributs que vous ne spécifiez pas explicitement à partir de la rubrique d'administration par défaut, appelée SYSTEM.DEFAULT.TOPIC. Pour voir quels sont ces attributs par défaut, utilisez la commande suivante:

```
DISPLAY TOPIC (SYSTEM.DEFAULT.TOPIC)
```

### Référence associée

[DISPLAY TOPIC](#)

[DISPLAY TPSTATUS](#)

## Modification des attributs de rubrique d'administration

Vous pouvez modifier les attributs de rubrique de deux manières, à l'aide de la commande **ALTER TOPIC** ou de la commande **DEFINE TOPIC** avec l'attribut **REPLACE**.

### Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

Par exemple, si vous souhaitez modifier la priorité par défaut des messages distribués à une rubrique appelée ORANGE.TOPIC, à la valeur 5, utilisez l'une des commandes suivantes:

- A l'aide de la commande **ALTER** :

```
ALTER TOPIC(ORANGE.TOPIC) DEFPRTY(5)
```

Cette commande modifie à 5 un attribut unique, celui de la priorité par défaut des messages distribués à cette rubrique ; tous les autres attributs restent identiques.

- A l'aide de la commande **DEFINE** :

```
DEFINE TOPIC(ORANGE.TOPIC) DEFPRTY(5) REPLACE
```

Cette commande modifie la priorité par défaut des messages distribués à cette rubrique. Tous les autres attributs reçoivent leurs valeurs par défaut.

Si vous modifiez la priorité des messages envoyés à cette rubrique, les messages existants ne sont pas affectés. Tout nouveau message, cependant, utilise la priorité spécifiée si elle n'est pas fournie par l'application de publication.

### Référence associée

[ALTER TOPIC](#)

[DISPLAY TOPIC](#)

## Copie d'une définition de rubrique d'administration

Vous pouvez copier une définition de rubrique à l'aide de l'attribut LIKE de la commande **DEFINE**.

### Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

La commande suivante crée une rubrique, MAGENTA.TOPIC, avec les mêmes attributs que la rubrique d'origine, ORANGE.TOPIC, plutôt que ceux de la rubrique d'administration par défaut du système. Entrez

le nom de la rubrique à copier exactement tel qu'il a été entré lors de la création de la rubrique. Si le nom contient des caractères minuscules, placez-le entre apostrophes.

```
DEFINE TOPIC (MAGENTA.TOPIC) +  
LIKE (ORANGE.TOPIC)
```

Vous pouvez également utiliser cette forme de la commande **DEFINE** pour copier une définition de rubrique, mais apporter des modifications aux attributs de l'original. Exemple :

```
DEFINE TOPIC (BLUE.TOPIC) +  
TOPICSTR (BLUE) +  
LIKE (ORANGE.TOPIC)
```

Vous pouvez également copier les attributs de la rubrique BLUE.TOPIC à la rubrique GREEN.TOPIC et indiquez que lorsque les publications ne peuvent pas être distribuées à leur file d'attente de souscription correcte, elles ne sont pas placées dans la file d'attente de rebut. Exemple :

```
DEFINE TOPIC (GREEN.TOPIC) +  
TOPICSTR (GREEN) +  
LIKE (BLUE.TOPIC) +  
USEDLQ (NO)
```

### Référence associée

[DEFINE TOPIC](#)

## Suppression d'une définition de rubrique d'administration

Vous pouvez utiliser la commande MQSC **DELETE TOPIC** pour supprimer une rubrique d'administration.

### Avant de commencer

**Remarque :** L'exemple de cette tâche nécessite que vous exécutiez des commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

```
DELETE TOPIC (ORANGE.TOPIC)
```

Les applications ne pourront plus ouvrir la rubrique pour la publication ou créer de nouveaux abonnements à l'aide du nom d'objet ORANGE.TOPIC. Les applications de publication pour lesquelles la rubrique est ouverte peuvent continuer à publier la chaîne de rubrique résolue. Tous les abonnements déjà effectués à cette rubrique continuent de recevoir des publications après la suppression de la rubrique.

Les applications qui ne font pas référence à cet objet de rubrique mais qui utilisent la chaîne de rubrique résolue représentée par cet objet de rubrique, 'ORANGE' dans cet exemple, continuent de fonctionner. Dans ce cas, ils héritent des propriétés d'un objet de rubrique plus haut dans l'arborescence de rubriques. Pour plus d'informations, voir [Arborescences de rubriques](#).

### Référence associée

[Supprimer la rubrique](#)

## Utilisation des abonnements

Utilisez les commandes MQSC pour gérer les abonnements.

## Pourquoi et quand exécuter cette tâche

Les abonnements peuvent être de l'un des trois types définis dans l'attribut **SUBTYPE** :

### ADMIN

Défini administrativement par un utilisateur.

### PROXY

Abonnement créé en interne pour le routage des publications entre les gestionnaires de files d'attente.

### API

Créé à l'aide d'un programme, par exemple, à l'aide de l'appel MQI MQSUB.

Pour plus d'informations sur ces commandes, voir [Commandes MQSC](#) .

## Définition d'un abonnement d'administration

Utilisez la commande MQSC **DEFINE SUB** pour créer un abonnement d'administration. Vous pouvez également utiliser la valeur par défaut définie dans la définition d'abonnement local par défaut. Vous pouvez également modifier les caractéristiques de l'abonnement à partir de celles de l'abonnement local par défaut, SYSTEM.DEFAULT.SUB créée lors de l'installation du système.

## Avant de commencer

**Remarque** : Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

La commande **DEFINE SUB** qui suit définit un abonnement appelé ORANGE avec les caractéristiques suivantes:

- Abonnement durable, c'est-à-dire qu'il est conservé après le redémarrage du gestionnaire de files d'attente, avec un délai d'expiration illimité.
- Réception des publications effectuées sur la chaîne de rubrique ORANGE, avec les priorités de message définies par les applications de publication.
- Les publications fournies pour cet abonnement sont envoyées à la file d'attente locale SUBQ. Cette file d'attente doit être définie avant la définition de l'abonnement.

```
DEFINE SUB (ORANGE) +  
TOPICSTR (ORANGE) +  
DESTCLAS (PROVIDED) +  
DEST (SUBQ) +  
EXPIRY (UNLIMITED) +  
PUBPRTY (ASPUB)
```

### Remarque :

- L'abonnement et le nom de la chaîne de rubrique ne doivent pas nécessairement correspondre.
- A l'exception des valeurs de la destination et de la chaîne de rubrique, toutes les valeurs d'attribut affichées sont les valeurs par défaut. Ils ne sont présentés ici qu'à titre d'illustration. Vous pouvez les omettre si vous êtes sûr que les valeurs par défaut correspondent à celles que vous souhaitez ou qu'elles n'ont pas été modifiées. Voir aussi [«Affichage des attributs des abonnements»](#), à la page 189.
- Si vous disposez déjà d'un abonnement local sur le même gestionnaire de files d'attente avec le nom ORANGE, cette commande échoue. Utilisez l'attribut **REPLACE** si vous souhaitez remplacer la définition existante d'une file d'attente, mais voir aussi [«Modification des attributs d'abonnement local»](#), à la page 190.
- Si la file d'attente SUBQ n'existe pas, cette commande échoue.

## Référence associée

[DEFINE SUB](#)

## Affichage des attributs des abonnements

Vous pouvez utiliser la commande **DISPLAY SUB** pour afficher les attributs configurés de tout abonnement connu du gestionnaire de files d'attente.

### Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

```
DISPLAY SUB(ORANGE)
```

Vous pouvez afficher les attributs de manière sélective en les spécifiant individuellement. Exemple :

```
DISPLAY SUB(ORANGE) +  
SUBID +  
TOPICSTR +  
DURABLE
```

Cette commande affiche les trois attributs spécifiés comme suit:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

TOPICSTR est la chaîne de rubrique résolue sur laquelle cet abonné fonctionne. Lorsqu'un abonnement est défini pour utiliser un objet de rubrique, la chaîne de rubrique de cet objet est utilisée comme préfixe de la chaîne de rubrique fournie lors de la création de l'abonnement. SUBID est un identificateur unique attribué par le gestionnaire de files d'attente lors de la création d'un abonnement. Il s'agit d'un attribut utile à afficher car certains noms d'abonnement peuvent être longs ou dans des jeux de caractères différents pour lesquels il peut devenir impossible de les afficher.

Une autre méthode d'affichage des abonnements consiste à utiliser le SUBID:

```
DISPLAY SUB +  
SUBID(414D51204141412020202020202020EE921E4E20002A03) +  
TOPICSTR +  
DURABLE
```

Cette commande génère la même sortie que précédemment:

```
AMQ8096: IBM MQ subscription inquired.  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
SUB(ORANGE) TOPICSTR(ORANGE)  
DURABLE(YES)
```

Les abonnements de proxy sur un gestionnaire de files d'attente ne sont pas affichés par défaut. Pour les afficher, spécifiez un **SUBTYPE** de type PROXY ou ALL.

Vous pouvez utiliser la commande `DISPLAY SBSTATUS` pour afficher les attributs d'exécution. Par exemple, utilisez la commande suivante:

```
DISPLAY SBSTATUS(ORANGE) NUMMSGS
```

La sortie suivante s'affiche :

```
AMQ8099: IBM MQ subscription status inquired.  
SUB(ORANGE)  
SUBID(414D51204141412020202020202020EE921E4E20002A03)  
NUMMSGS(0)
```

Lorsque vous définissez un abonnement d'administration, il prend tous les attributs que vous ne spécifiez pas explicitement à partir de l'abonnement par défaut, appelé `SYSTEM.DEFAULT.SUB`. Pour voir quels sont ces attributs par défaut, utilisez la commande suivante:

```
DISPLAY SUB (SYSTEM.DEFAULT.SUB)
```

### Référence associée

[DISPLAY SUB](#)

## Modification des attributs d'abonnement local

Vous pouvez modifier les attributs d'abonnement de deux manières, à l'aide de la commande **ALTER SUB** ou de la commande **DEFINE SUB** avec l'attribut **REPLACE**.

### Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

Si vous souhaitez modifier la priorité des messages distribués à un abonnement appelé `ORANGE` sur 5, utilisez l'une des commandes suivantes:

- A l'aide de la commande **ALTER** :

```
ALTER SUB(ORANGE) PUBPRTY(5)
```

Cette commande modifie un attribut unique, celui de la priorité des messages distribués à cet abonnement à 5 ; tous les autres attributs restent les mêmes.

- A l'aide de la commande **DEFINE** :

```
DEFINE SUB(ORANGE) PUBPRTY(5) REPLACE
```

Cette commande modifie non seulement la priorité des messages distribués à cet abonnement, mais également tous les autres attributs auxquels sont attribuées leurs valeurs par défaut.

Si vous modifiez la priorité des messages envoyés à cet abonnement, les messages existants ne sont pas affectés. Toutefois, tous les nouveaux messages ont la priorité spécifiée.

### Référence associée

[ALTER SUB](#)

[DEFINE SUB](#)

## Copie d'une définition d'abonnement local

Vous pouvez copier une définition d'abonnement à l'aide de l'attribut **LIKE** de la commande **DEFINE**.

### Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

```
DEFINE SUB(BLUE) +  
LIKE(ORANGE)
```

Vous pouvez également copier les attributs de la sous-commande REAL dans la sous-commande THIRD.SUB et indiquer que le **correlID** des publications distribuées est THIRD, plutôt que les diffuseurs de publications **correlID**. Exemple :

```
DEFINE SUB(THIRD.SUB) +  
LIKE(BLUE) +  
DETCORL(ORANGE)
```

### Référence associée

[DEFINE SUB](#)

## Suppression d'un abonnement local

Vous pouvez utiliser la commande MQSC **DELETE SUB** pour supprimer un abonnement local.

### Avant de commencer

**Remarque :** Les exemples de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Exemple

```
DELETE SUB(ORANGE)
```

Vous pouvez également supprimer un abonnement à l'aide du SUBID:

```
DELETE SUB SUBID(414D51204141412020202020202020EE921E4E20002A03)
```

### Référence associée

[SUPPRIMER DES SOUS](#)

## Vérification des messages sur un abonnement

Lorsqu'un abonnement est défini, il est associé à une file d'attente. Les messages publiés correspondant à cet abonnement sont placés dans cette file d'attente. Utilisez les commandes MQSC pour rechercher les messages actuellement mis en file d'attente pour un abonnement.

## Avant de commencer

**Remarque :** Les étapes de cette tâche nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

## Pourquoi et quand exécuter cette tâche

Notez que les commandes MQSC suivantes affichent uniquement les abonnements qui ont reçu des messages.

Pour rechercher les messages actuellement mis en file d'attente pour un abonnement, procédez comme suit:

### Procédure

1. Pour rechercher les messages mis en file d'attente pour un type d'abonnement DISPLAY SBSTATUS (*sub\_name*) NUMMSGs, voir «Affichage des attributs des abonnements», à la page 189.
2. Si la valeur **NUMMSGs** est supérieure à zéro, identifiez la file d'attente associée à l'abonnement en entrant DISPLAY SUB (*sub\_name*) DEST.
3. En utilisant le nom de la file d'attente renvoyée, vous pouvez afficher les messages en suivant la technique décrite dans «Exploration des files d'attente avec l'exemple de programme», à la page 151.

### Référence associée

[STATUT DU JEU DE CARACTÈRES D'AFFICHAGE](#)

## Utilisation des services

Les objets de service sont un moyen par lequel des processus supplémentaires peuvent être gérés dans le cadre d'un gestionnaire de files d'attente. Avec les services, vous pouvez définir des programmes qui sont démarrés et arrêtés lorsque le gestionnaire de files d'attente démarre et se termine. Les services IBM MQ sont toujours démarrés sous l'ID de l'utilisateur qui a démarré le gestionnaire de files d'attente.

## Pourquoi et quand exécuter cette tâche

Les objets de service peuvent être de l'un des types suivants:

### serveur

Un serveur est un objet de service dont le paramètre **SERVTYPE** est spécifié comme SERVER. Un objet de service serveur est la définition d'un programme qui est exécuté lorsqu'un gestionnaire de files d'attente spécifié est démarré. Les objets de service serveur définissent des programmes qui s'exécutent généralement pendant une longue période. Par exemple, un objet de service de serveur peut être utilisé pour exécuter un processus de moniteur de déclenchement, tel que **runmqtrm**.

Une seule instance d'un objet de service de serveur peut s'exécuter simultanément. Le statut des objets de service de serveur en cours d'exécution peut être surveillé à l'aide de la commande MQSC, **DISPLAY SVSTATUS**.

### Commande

Une commande est un objet de service dont le paramètre **SERVTYPE** est spécifié sous la forme COMMAND. Les objets de service de commande sont similaires aux objets de service de serveur, mais plusieurs instances d'un objet de service de commande peuvent s'exécuter simultanément et leur statut ne peut pas être surveillé à l'aide de la commande MQSC **DISPLAY SVSTATUS**.

Si la commande MQSC, **STOP SERVICE**, est exécutée, aucune vérification n'est effectuée pour déterminer si le programme démarré par la commande MQSC, **START SERVICE**, est toujours actif avant l'arrêt du programme.

### Référence associée

[Définir un service](#)

[STATUT DE L'AFFICHAGE](#)



Démarrage du service

Arrêt du service

## Définition d'un objet de service

Vous définissez un objet de service à l'aide de la commande MQSC **DEFINE SERVICE**.

### Avant de commencer

**Remarque :** Cette tâche nécessite l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

### Procédure

- Définissez un objet de service à l'aide de la commande MQSC **DEFINE SERVICE**.

Les attributs que vous devez définir sont les suivants:

#### **SERVTYPE**

Définit le type de l'objet de service. Les valeurs admises sont les suivantes :

##### **SERVEUR**

Un objet de service de serveur.

Une seule instance d'un objet de service de serveur peut être exécutée à la fois. Le statut des objets de service du serveur peut être surveillé à l'aide de la commande MQSC, **DISPLAY SVSTATUS**.

##### **Commande**

Objet de service de commande.

Plusieurs instances d'un objet de service de commande peuvent être exécutées simultanément. L'état des objets de service de commande ne peut pas être surveillé.

#### **STARTCMD**

Programme exécuté pour démarrer le service. Vous devez indiquer un chemin d'accès complet au programme.

#### **STARTARG**

Arguments transmis au programme de démarrage.

#### **STDERR**

Indique le chemin d'accès à un fichier vers lequel l'erreur standard (stderr) du programme de service doit être redirigée.

#### **STDOUT**

Indique le chemin d'accès à un fichier vers lequel la sortie standard (stdout) du programme de service doit être redirigée.

#### **STOPCMD**

Programme exécuté pour arrêter le service. Vous devez indiquer un chemin d'accès complet au programme.

#### **STOPARG**

Arguments transmis au programme d'arrêt.

#### **CONTROL**

Indique comment le service doit être démarré et arrêté:

##### **MANUAL**

Le service ne doit pas être démarré automatiquement ou arrêté automatiquement. Il est contrôlé par l'utilisation des commandes **START SERVICE** et **STOP SERVICE** . Il s'agit de la valeur par défaut.

##### **QMGR**

Le service défini doit être démarré et arrêté en même temps que le gestionnaire de files d'attente.

## STARTONLY (UNIQUEMENT)

Le service doit être démarré en même temps que le gestionnaire de files d'attente, mais il n'est pas demandé de s'arrêter lorsque le gestionnaire de files d'attente est arrêté.

### Tâches associées

«Gestion des services», à la page 194

Une instance d'un objet de service peut être démarrée et arrêtée automatiquement par le gestionnaire de files d'attente ou démarrée et arrêtée à l'aide des commandes MQSC **START SERVICE** et **STOP SERVICE**.

### Référence associée

[Définir un service](#)

[STATUT DE L'AFFICHAGE](#)

[Démarrage du service](#)

[Arrêt du service](#)

## Gestion des services

Une instance d'un objet de service peut être démarrée et arrêtée automatiquement par le gestionnaire de files d'attente ou démarrée et arrêtée à l'aide des commandes MQSC **START SERVICE** et **STOP SERVICE**.

## Avant de commencer

**Remarque :** Cette tâche nécessite l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

## Procédure

- Définissez le paramètre **CONTROL** sur le gestionnaire de files d'attente pour démarrer ou arrêter automatiquement une instance d'un objet de service, ou utilisez les commandes MQSC **START SERVICE** et **STOP SERVICE** pour le faire manuellement.

Lorsqu'une instance d'un objet de service est démarrée, un message contenant le nom de l'objet de service et l'ID de processus du processus démarré est consigné dans le journal des erreurs du gestionnaire de files d'attente. Voici un exemple d'entrée de journal pour le démarrage d'un objet de service de serveur:

```
02/15/2005 11:54:24 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5028: The Server 'S1' has started. ProcessId(13031).

EXPLANATION:
The Server process has started.
ACTION:
None.
```

Voici un exemple d'entrée de journal pour le démarrage d'un objet de service de commande:

```
02/15/2005 11:53:55 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5030: The Command 'C1' has started. ProcessId(13030).

EXPLANATION:
The Command has started.
ACTION:
None.
```

Lorsqu'un service de serveur d'instance s'arrête, un message est consigné dans les journaux d'erreurs du gestionnaire de files d'attente contenant le nom du service et l'ID de processus du processus en cours de fin. Voici un exemple d'entrée de journal pour un objet de service de serveur en cours d'arrêt:

```
02/15/2005 11:54:54 AM - Process(10363.1) User(mqm) Program(amqzmgr0)
Host(HOST_1) Installation(Installation1)
VRMF(7.1.0.0) QMgr(A.B.C)
AMQ5029: The Server 'S1' has ended. ProcessId(13031).
```

```
EXPLANATION:
The Server process has ended.
ACTION:
None.
```

### Tâches associées

«Définition de variables d'environnement supplémentaires dans le fichier `service.env`», à la page 195  
L'environnement dans lequel le processus d'un service est démarré est hérité de celui du gestionnaire de files d'attente. Il est possible de définir des variables d'environnement supplémentaires à définir dans l'environnement du processus de service en ajoutant les variables à définir à l'un des fichiers de substitution d'environnement `service.env`.

### Référence associée

[STOP SERVICE \(arrêt d'un service\) sur Multiplatforms](#)

[START SERVICE \(démarrage d'un service\) sur Multiplatforms](#)

## Définition de variables d'environnement supplémentaires dans le fichier `service.env`

L'environnement dans lequel le processus d'un service est démarré est hérité de celui du gestionnaire de files d'attente. Il est possible de définir des variables d'environnement supplémentaires à définir dans l'environnement du processus de service en ajoutant les variables à définir à l'un des fichiers de substitution d'environnement `service.env`.

### Pourquoi et quand exécuter cette tâche

Il existe deux fichiers possibles auxquels vous pouvez ajouter des variables d'environnement:

- Le fichier `service.env` de la portée de la machine
- Fichier `service.env` de portée du gestionnaire de files d'attente

Les deux fichiers sont traités, s'ils sont disponibles, avec les définitions du fichier de portée du gestionnaire de files d'attente prioritaires sur celles du fichier de portée de la machine.

Vous pouvez spécifier n'importe quelle variable d'environnement dans le fichier `service.env`. Par exemple, si le service IBM MQ exécute un certain nombre de commandes, il peut être utile de définir la variable utilisateur **PATH** dans le fichier `service.env`.

**Remarque :** Les valeurs que vous définissez pour la variable ne peuvent pas être des variables d'environnement ; par exemple, `CLASSPATH= %CLASSPATH%` est incorrect. De même, sur Linux `PATH=$PATH :/opt/mqm/bin`, des résultats inattendus sont générés.


**CLASSPATH** doit être en majuscules et l'instruction de chemin d'accès aux classes ne peut contenir que des littéraux. Certains services (télémétrie, par exemple) définissent leur propre chemin d'accès aux classes. Le **CLASSPATH** défini dans `service.env` lui est ajouté.

Le format des variables définies dans le fichier `service.env` est une liste de paires de variables de nom et de valeur. Chaque variable doit être définie sur une nouvelle ligne, et chaque variable est prise telle qu'elle est explicitement définie, y compris les espaces.

### Procédure

- Ajoutez des variables d'environnement au fichier `service.env` de la portée de la machine.

Ce fichier se trouve dans:

–  `/var/mqm` sur les systèmes AIX and Linux.

- **Windows** Répertoire de données sélectionné lors de l'installation sur les systèmes Windows .
- Ajoutez des variables d'environnement au fichier `service.env` de portée du gestionnaire de files d'attente.  
Ce fichier se trouve dans le répertoire de données du gestionnaire de files d'attente. Par exemple, l'emplacement du fichier de substitution d'environnement pour un gestionnaire de files d'attente nommé QMNAME est:
  - **Linux** **AIX** Sur les systèmes AIX and Linux , `/var/mqm/qmgrs/QMNAME/service.env`
  - **Windows** Sur les systèmes Windows ,  
`C:\ProgramData\IBM\MQ\qmgrs\QMNAME\service.env`

### Exemple de fichier `service.env`

```

#*****#
#*                                     *#
#* <N_OCO_COPYRIGHT>                 *#
#* Licensed Materials - Property of IBM *#
#*                                     *#
#* 63H9336                             *#
#* (C) Copyright IBM Corporation 2005, 2024. *#
#*                                     *#
#* <NOC_COPYRIGHT>                   *#
#*                                     *#
#*****#
#* Module Name: service.env           *#
#* Type          : IBM MQ service environment file *#
#* Function      : Define additional environment variables to be set *#
#*               : for SERVICE programs. *#
#* Usage        : <VARIABLE>=<VALUE> *#
#*               *#
#*****#
MYLOC=/opt/myloc/bin
MYTMP=/tmp
TRACEDIR=/tmp/trace
MYINITQ=ACCOUNTS.INITIATION.QUEUE

```

### Tâches associées

«Utilisation d'insertions remplaçables sur des définitions de service», à la page 196

Vous pouvez remplacer des jetons dans la définition d'un objet de service. Les jetons qui sont remplacés sont automatiquement remplacés par leur texte développé lors de l'exécution du programme de service.

### Référence associée

[Description des variables d'environnement](#)

## Utilisation d'insertions remplaçables sur des définitions de service

Vous pouvez remplacer des jetons dans la définition d'un objet de service. Les jetons qui sont remplacés sont automatiquement remplacés par leur texte développé lors de l'exécution du programme de service.

### Pourquoi et quand exécuter cette tâche

Les jetons de substitution peuvent être extraits de la liste suivante de jetons communs ou de toute variable définie dans le fichier `service.env`.

### Procédure

- Pour utiliser des insertions remplaçables, insérez le jeton dans les caractères + dans les chaînes **STARTCMD**, **STARTARG**, **STOPCMD**, **STOPARG**, **STDOUT** ou **STDERR** .

Pour des exemples, voir «Utilisation d'un objet de service de serveur», à la page 197 et «Utilisation d'un objet de service de commande», à la page 199.



Les jetons communs suivants peuvent être utilisés pour remplacer des jetons dans la définition d'un objet de service:

#### **CHEMIN\_INSTALL\_MQ**

Emplacement dans lequel IBM MQ est installé.

#### **MQ\_DATA\_PATH**

Emplacement du répertoire de données IBM MQ :

-  Sur les systèmes AIX and Linux , l'emplacement du répertoire de données IBM MQ est /var/mqm/
-  Sur les systèmes Windows , l'emplacement du répertoire de données IBM MQ est le répertoire de données sélectionné lors de l'installation de IBM MQ

#### **QMNAME**

Nom du gestionnaire de files d'attente en cours.

#### **NOM\_SERVICE\_MQ**

Nom du service.

#### **PID SERVEUR\_MQ**

Ce jeton ne peut être utilisé que par les arguments **STOPARG** et **STOPCMD** .

Pour les objets de service de serveur, ce jeton est remplacé par l'ID de processus du processus démarré par les arguments **STARTCMD** et **STARTARG** . Sinon, ce jeton est remplacé par 0.

#### **CHEMIN\_DATA\_MQ\_Q\_MGR\_**

Emplacement du répertoire de données du gestionnaire de files d'attente.

#### **MQ\_Q\_MGR\_DATA\_NAME**

Nom transformé du gestionnaire de files d'attente. Pour plus d'informations sur la transformation de nom, voir [Description des noms de fichier IBM MQ](#).

## Utilisation d'un objet de service de serveur

Ces exemples montrent comment définir, utiliser et modifier un objet de service de serveur pour démarrer un moniteur de déclenchement ou un autre programme.

### Avant de commencer

**Remarque :** Ces exemples nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

Ces exemples sont écrits avec des caractères de séparation de chemin de style UNIX , sauf indication contraire.

### Procédure

#### 1. Définissez un objet de service de serveur à l'aide de la commande **DEFINE SERVICE MQSC**:

```
DEFINE SERVICE(S1) +  
CONTROL(QMGR) +  
SERVTYPE(SERVER) +  
STARTCMD('+MQ_INSTALL_PATH+bin/runmqtrm') +  
STARTARG('-m +QMNAME+ -q ACCOUNTS.INITIATION.QUEUE') +  
STOPCMD('+MQ_INSTALL_PATH+bin/amqsstop') +  
STOPARG('-m +QMNAME+ -p +MQ_SERVER_PID+')
```

où :

- +MQ\_INSTALL\_PATH+ est un jeton représentant le répertoire d'installation.
- +QMNAME+ est un jeton représentant le nom du gestionnaire de files d'attente.
- ACCOUNTS.INITIATION.QUEUE est la file d'attente d'initialisation.

amqsstop est un exemple de programme fourni avec IBM MQ qui demande au gestionnaire de files d'attente d'interrompre toutes les connexions pour l'ID de processus. amqsstop génère des commandes PCF. Par conséquent, le serveur de commandes doit être en cours d'exécution.

+MQ\_SERVER\_PID+ est un jeton représentant l'ID de processus transmis au programme d'arrêt.

Pour obtenir la liste des jetons communs, voir [«Utilisation d'insertions remplaçables sur des définitions de service»](#), à la page 196 .

2. **Une instance de l'objet de service serveur s'exécute lors du prochain démarrage du gestionnaire de files d'attente. Toutefois, vous pouvez démarrer une instance de l'objet de service serveur immédiatement à l'aide de la commande `START SERVICE MQSC`:**

```
START SERVICE(S1)
```

3. **Affichez le statut du processus de service du serveur à l'aide de la commande `DISPLAY SVSTATUS MQSC`:**

```
DISPLAY SVSTATUS(S1)
```

4. **Modifier l'objet de service du serveur et prendre en compte les mises à jour en redémarrant manuellement le processus de service du serveur à l'aide de la commande `ALTER SERVICE MQSC`.**

L'objet de service du serveur est modifié de sorte que la file d'attente d'initialisation soit spécifiée sous la forme JUPITER.INITIATION.QUEUE.

```
ALTER SERVICE(S1) +  
STARTARG(' -m +QMNAME+ -q JUPITER.INITIATION.QUEUE')
```

**Remarque :** Un service en cours d'exécution ne récupère aucune mise à jour de sa définition de service tant qu'il n'est pas redémarré.

5. **Redémarrez le processus de service du serveur afin que les modifications soient prises en compte, à l'aide des commandes `STOP SERVICE` et `START SERVICE MQSC`:**

```
STOP SERVICE(S1)
```

suivie de :

```
START SERVICE(S1)
```

Le processus de service du serveur est redémarré et prend en compte les modifications apportées dans «4», à la page 198.

**Remarque :** La commande MQSC, `STOP SERVICE`, ne peut être utilisée que si un argument `STOPCMD` est spécifié dans la définition de service.

### Autres exemples de transmission d'arguments

- **Définissez un objet de service serveur pour démarrer un programme appelé `runserv` lorsqu'un gestionnaire de files d'attente est démarré.**

Pour ce faire, utilisez la commande `DEFINE SERVICE MQSC`.

Cet exemple est écrit avec des caractères de séparation de chemin de style Windows .

L'un des arguments transmis au programme de démarrage est une chaîne contenant un espace. Cet argument doit être transmis sous la forme d'une chaîne unique. Pour ce faire, des guillemets sont utilisés comme indiqué dans la commande suivante pour définir l'objet de service de commande.

```
DEFINE SERVICE(S1) SERVTYPE(SERVER) CONTROL(QMGR) +
```

```
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

```
DEFINE SERVICE(S4) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('C:\Program Files\Tools\runserv.exe') +
STARTARG('-m +QMNAME+ -d "C:\Program Files\Tools\'') +
STDOUT('C:\Program Files\Tools\+MQ_SERVICE_NAME+.out')
```

où :

+QMNAME+ est un jeton représentant le nom du gestionnaire de files d'attente.

"C:\Program Files\Tools\'" est une chaîne contenant un espace, qui sera transmise sous la forme d'une chaîne unique.

- **Définissez un objet de service serveur pouvant être utilisé pour démarrer automatiquement le moniteur de déclenchement au démarrage du gestionnaire de files d'attente.**

Pour ce faire, utilisez la commande **DEFINE SERVICE MQSC**.

```
DEFINE SERVICE(TRIG_MON_START) +
CONTROL(QMGR) +
SERVTYPE(SERVER) +
STARTCMD('runmqtrm') +
STARTARG('-m +QMNAME+ -q +IQNAME+')
```

où :

+QMNAME+ est un jeton représentant le nom du gestionnaire de files d'attente.

+IQNAME+ est une variable d'environnement définie par l'utilisateur dans l'un des fichiers service.env représentant le nom de la file d'attente d'initialisation.

### Référence associée

[MODIFIER LE SERVICE](#)

[Définir un service](#)

[STATUT DE L'AFFICHAGE](#)

[Démarrage du service](#)

[Arrêt du service](#)

## Utilisation d'un objet de service de commande

Ces exemples montrent comment définir un objet de service de commande pour démarrer un programme qui écrit des entrées dans le journal système du système d'exploitation lorsqu'un gestionnaire de files d'attente est démarré ou arrêté.

### Avant de commencer

**Remarque :** Ces exemples nécessitent l'exécution de la commande **DEFINE SERVICE MQSC**. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

Ces exemples sont écrits avec des caractères de séparation de chemin de style UNIX .

### Pourquoi et quand exécuter cette tâche

Dans les exemples suivants:

Logger est un exemple de programme fourni avec IBM MQ qui peut écrire des entrées dans le journal système du système d'exploitation.

+QMNAME+ est un jeton représentant le nom du gestionnaire de files d'attente.

## Procédure

- Définissez un objet de service de commande pour démarrer un programme qui écrit des entrées dans le journal système du système d'exploitation lorsqu'un gestionnaire de files d'attente est démarré ou arrêté:

```
DEFINE SERVICE(S2) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STARTCMD('/usr/bin/logger') +
STARTARG('Queue manager +QMNAME+ starting') +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

- Définissez un objet de service de commande pour démarrer un programme qui écrit des entrées dans le journal système du système d'exploitation uniquement lorsqu'un gestionnaire de files d'attente est arrêté:

```
DEFINE SERVICE(S3) +
CONTROL(QMGR) +
SERVTYPE(COMMAND) +
STOPCMD('/usr/bin/logger') +
STOPARG('Queue manager +QMNAME+ stopping')
```

### Référence associée

[Définir un service](#)

## Gestion des objets pour le déclenchement

Ces exemples montrent comment démarrer automatiquement une application lorsque certaines conditions d'une file d'attente sont remplies. Par exemple, vous pouvez démarrer une application lorsque le nombre de messages d'une file d'attente atteint un nombre spécifié. Cette fonction est appelée *déclenchement*. Vous devez définir les objets qui prennent en charge le déclenchement.

### Avant de commencer

**Remarque :** Ces exemples nécessitent l'exécution de commandes MQSC. La façon dont vous effectuez cette opération varie en fonction de la plateforme. Voir [Administration d' IBM MQ à l'aide de commandes MQSC](#).

Ces exemples sont écrits avec des caractères de séparation de chemin de style UNIX .

### Pourquoi et quand exécuter cette tâche

Pour une description détaillée du déclenchement, voir [Démarrage des applications IBM MQ à l'aide de déclencheurs](#).

## Procédure

- Définissez une file d'attente d'application pour le déclenchement.

Une file d'attente d'application est une file d'attente locale utilisée par les applications pour la messagerie, via l'interface MQI. Le déclenchement requiert la définition d'un certain nombre d'attributs de file d'attente dans la file d'attente d'application.

Le déclenchement lui-même est activé par l'attribut **Trigger** (TRIGGER dans les commandes MQSC). Dans cet exemple, un événement déclencheur doit être généré lorsqu'il y a 100 messages de priorité 5 ou supérieure dans la file d'attente locale MOTOR.INSURANCE.QUEUE, comme suit:

```
DEFINE QLOCAL (MOTOR.INSURANCE.QUEUE) +
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +
MAXMSGL (2000) +
DEFPSIST (YES) +
```



```
INITQ (MOTOR.INS.INIT.QUEUE) +  
TRIGGER +  
TRIGTYPE (DEPTH) +  
TRIGDPTH (100)+  
TRIGMPRI (5)
```

où :

**QLOCAL (MOTOR.INSURANCE.QUEUE)**

Nom de la file d'attente d'application en cours de définition.

**PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)**

Nom de la définition de processus qui définit l'application à démarrer par un programme de moniteur de déclenchement.

**MAXMSGL (2000)**

Longueur maximale des messages dans la file d'attente.

**DEFPSIST (YES)**

Indique que les messages de cette file d'attente sont persistants par défaut.

**INITQ (MOTOR.INS.INIT.QUEUE)**

Nom de la file d'attente d'initialisation dans laquelle le gestionnaire de files d'attente doit insérer le message de déclenchement.

**TRIGGER**

Valeur de l'attribut de déclencheur.

**TRIGTYPE (DEPTH)**

Indique qu'un événement déclencheur est généré lorsque le nombre de messages de priorité requise (TRIGMPRI) atteint le nombre indiqué dans TRIGDPTH.

**TRIGDPTH (100)**

Nombre de messages requis pour générer un événement déclencheur.

**TRIGMPRI (5)**

Priorité des messages qui doivent être comptés par le gestionnaire de files d'attente pour déterminer s'il convient de générer un événement déclencheur. Seuls les messages de priorité 5 ou supérieure sont comptés.

- Définition d'une file d'attente d'initialisation

Lorsqu'un événement déclencheur se produit, le gestionnaire de files d'attente place un message de déclenchement dans la file d'attente d'initialisation spécifiée dans la définition de file d'attente d'application. Les files d'attente d'initialisation n'ont pas de paramètres spéciaux, mais vous pouvez utiliser la définition suivante de la file d'attente locale MOTOR.INS.INIT.QUEUE pour obtenir des conseils:

```
DEFINE QLOCAL (MOTOR.INS.INIT.QUEUE) +  
GET (ENABLED) +  
NOSHARE +  
NOTRIGGER +  
MAXMSGL (2000) +  
MAXDEPTH (1000)
```

- Définir un processus

Utilisez la commande DEFINE PROCESS pour créer une définition de processus. Une définition de processus définit l'application à utiliser pour traiter les messages de la file d'attente d'application. La définition de file d'attente d'application nomme le processus à utiliser et associe ainsi la file d'attente d'application à l'application à utiliser pour traiter ses messages. Cette opération est effectuée via l'attribut PROCESS de la file d'attente d'application MOTOR.INSURANCE.QUEUE. La commande MQSC suivante définit le processus requis, MOTOR.INSURANCE.QUOTE.PROCESS, identifié dans cet exemple:

```
DEFINE PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) +  
DESCR ('Insurance request message processing') +  
APPLTYPE (UNIX) +
```

```
APPLICID ('/u/admin/test/IRMP01') +
USERDATA ('open, close, 235')
```

où :

#### **MOTOR.INSURANCE.QUOTE.PROCESS**

Nom de la définition de processus.

#### **DESCR ('Insurance request message processing')**

Décrit le programme d'application auquel cette définition se rapporte. Ce texte s'affiche lorsque vous utilisez la commande DISPLAY PROCESS. Cela peut vous aider à identifier ce que le processus fait. Si vous utilisez des espaces dans la chaîne, vous devez placer la chaîne entre apostrophes.

#### **APPLTYPE (UNIX)**

Type d'application à démarrer.

#### **APPLICID ('/u/admin/test/IRMP01')**

Nom du fichier exécutable de l'application, indiqué comme nom de fichier qualifié complet. Sur les systèmes Windows, une valeur APPLICID typique serait c:\app1\test\irmp01.exe.

#### **USERDATA ('open, close, 235')**

Données définies par l'utilisateur qui peuvent être utilisées par l'application.

- Afficher les attributs d'une définition de processus

Utilisez la commande DISPLAY PROCESS pour examiner les résultats de votre définition. Exemple :

```
DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)

24 : DISPLAY PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS) ALL
AMQ8407: Display Process details.
DESCR ('Insurance request message processing')
APPLICID ('/u/admin/test/IRMP01')
USERDATA (open, close, 235)
PROCESS (MOTOR.INSURANCE.QUOTE.PROCESS)
APPLTYPE (UNIX)
```

Vous pouvez également utiliser la commande MQSC ALTER PROCESS pour modifier une définition de processus existante et la commande DELETE PROCESS pour supprimer une définition de processus.

## Utilisation de l'utilitaire **dmpmqmsg** entre deux systèmes

L'utilitaire **dmpmqmsg** (anciennement *qload*) permet de copier ou de déplacer le contenu d'une file d'attente ou de ses messages dans un fichier.

### Présentation

Le fichier que vous créez avec **dmpmqmsg** peut être sauvegardé au besoin et utilisé ultérieurement pour recharger les messages dans la file d'attente.

#### Important :

1. Le fichier possède un format spécifique compris par l'utilitaire. Toutefois, le fichier est lisible par l'utilisateur, de sorte que vous pouvez le mettre à jour dans un éditeur avant de le recharger. Si vous éditez le fichier, vous ne devez pas modifier son format.
2. L'utilitaire **dmpmqmsg** est fourni avec l'ensemble de fichiers d'exécution pour AIX, Linux, and Windows, de sorte qu'il est disponible sur le serveur et le client IBM MQ .

Les utilisations possibles sont les suivantes:

- Enregistrement des messages d'une file d'attente dans un fichier. Possibilité à des fins d'archivage, puis rechargement dans une file d'attente.
- Rechargement d'une file d'attente avec les messages précédemment sauvegardés dans un fichier.
- Suppression des anciens messages d'une file d'attente.

- "Réexécution" des messages de test à partir d'un emplacement enregistré, en conservant même l'heure correcte entre les messages, le cas échéant.



**Avertissement :** SupportPac MO03 a utilisé le paramètre **-1** pour spécifier la liaison locale ou client. **-1** a été remplacé par le paramètre **-c** .

**-P** est désormais utilisé pour les informations de page de codes à la place de **-c** .

Pour plus d'informations sur la commande et les paramètres disponibles, voir [dmpmqmsg](#) .

## Exemple d'utilisation de l'utilitaire **dmpmqmsg** sous Linux, à l'aide d'une machine Windows

Vous disposez d'un gestionnaire de files d'attente sur une machine Linux comportant des messages dans une file d'attente (*Q1*) que vous souhaitez déplacer dans une autre file d'attente (*Q2*) dans le même gestionnaire de files d'attente. Vous souhaitez lancer l'utilitaire **dmpmqmsg** à partir d'une machine Windows .

La file d'attente (*Q1*) comporte quatre messages qui ont été ajoutés à l'aide de l'exemple d'application **amqspwt** (gestionnaire de files d'attente locales) ou **amqspwtc** (gestionnaire de files d'attente éloignées).

Sur la machine Linux , vous voyez:

```
display q1(Q1) CURDEPTH
      2 : display q1(Q1) CURDEPTH
AMQ8409: Display Queue details.
      QUEUE(Q1)
      TYPE(QLOCAL)
      CURDEPTH(4)
```

Définissez la variable d'environnement MQSERVER pour qu'elle pointe vers le gestionnaire de files d'attente dans Linux. Exemple :

```
set MQSERVER=SYSTEM.DEF.SVRCONN/TCP/veracruz.x.com(1414)
```

où *veracruz* est le nom de la machine.

Exécutez l'utilitaire **dmpmqmsg** pour lire la file d'attente, *Q1*, et stocker la sortie dans `c:\temp\mqqload.txt`.

Connectez-vous en tant que client distant au gestionnaire de files d'attente, *QM\_VER*, s'exécutant sur l'hôte Linux et le port établis par MQSERVER. Vous obtenez la connexion en tant que client distant à l'aide de l'attribut: `-c`.

```
dmpmqmsg -m QM_VER -i Q1 -f c:\temp\mqqload.txt -c
Read      - Files:      0  Messages:      4  Bytes:      22
Written   - Files:      1  Messages:      4  Bytes:      22
```

Le fichier de sortie `c:\temp\mqqload.txt` contient du texte, en utilisant un format compris par l'utilitaire **dmpmqmsg** .

Sur la machine Windows , exécutez la commande **dmpmqmsg** (à l'aide de l'option `-o` à la place de l'option `-i`) pour charger la file d'attente (*Q2*) sur la machine Linux à partir d'un fichier sur la machine Windows :

```
dmpmqmsg -m QM_VER -o Q2 -f c:\temp\mqqload.txt -c
Read      - Files:      1  Messages:      4  Bytes:      22
Written   - Files:      0  Messages:      4  Bytes:      22
```

Sur la machine Linux , notez que quatre messages de la file d'attente ont été restaurés à partir du fichier.

```
display q1(Q2) CURDEPTH
      6 : display q1(Q2) CURDEPTH
AMQ8409: Display Queue details.
```

```
QUEUE(Q2)
TYPE(QLOCAL)
CURDEPTH(4)
```

Sur la machine Linux ,

Supprimez les messages de la file d'attente d'origine.

```
clear qlocal(Q1)
4 : clear qlocal(Q1)
AMQ8022: IBM MQ queue cleared.
```

Confirmez qu'il n'y a plus de messages dans la file d'attente d'origine:

```
display ql(Q1) CURDEPTH
5 : display ql(Q1) CURDEPTH
AMQ8409: Display Queue details.
QUEUE(Q1)
TYPE(QLOCAL)
CURDEPTH(0)
```

Pour obtenir une description de la commande et de ses paramètres, voir [dmpmqmsg](#) .

### Concepts associés

«Exemples d'utilisation de l'utilitaire [dmpmqmsg](#)», à la page 204

Méthodes simples d'utilisation de l'utilitaire **dmpmqmsg** (anciennement **qload**).

## Exemples d'utilisation de l'utilitaire dmpmqmsg

Méthodes simples d'utilisation de l'utilitaire **dmpmqmsg** (anciennement **qload**).

### Déchargement d'une file d'attente dans un fichier

Utilisez les options suivantes sur la ligne de commande pour sauvegarder les messages qui se trouvent dans une file d'attente dans un fichier:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile
```

Cette commande prend une copie des messages de la file d'attente et les sauvegarde dans le fichier indiqué.

### Déchargement d'une file d'attente dans une série de fichiers

Vous pouvez télécharger une file d'attente dans une série de fichiers en utilisant le caractère `insert` dans le nom de fichier. Dans ce mode, chaque message est écrit dans un nouveau fichier:

```
dmpmqmsg -m QM1 -i Q1 -f c:\myfile%n
```

Cette commande décharge la file d'attente dans les fichiers `myfile1`, `myfile2`, `myfile3`, etc.

### Chargement d'une file d'attente à partir d'un fichier

Pour recharger une file d'attente avec les messages que vous avez sauvegardés dans «[Déchargement d'une file d'attente dans un fichier](#)», à la page 204, utilisez les options suivantes sur la ligne de commande:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Cette commande décharge la file d'attente dans les fichiers `myfile1`, `myfile2`, `myfile3`, etc.

## Chargement d'une file d'attente à partir d'une série de fichiers

Vous pouvez charger une file d'attente à partir d'une série de fichiers en utilisant le caractère `insert` dans le nom de fichier. Dans ce mode, chaque message est écrit dans un nouveau fichier:

```
dmpmqmsg -m QM1 -o Q1 -f c:\myfile%n
```

Cette commande charge la file d'attente dans les fichiers `myfile1`, `myfile2`, `myfile3`, etc.

## Copier les messages d'une file d'attente vers une autre file d'attente

Remplacez le paramètre de fichier dans [«Déchargement d'une file d'attente dans un fichier»](#), à la page 204 par un autre nom de file d'attente et utilisez les options suivantes:

```
dmpmqmsg -m QM1 -i Q1 -o Q2
```

Cette commande permet de copier les messages d'une file d'attente vers une autre.

## Copier les 100 premiers messages d'une file d'attente vers une autre

Utilisez la commande de l'exemple précédent et ajoutez l'option `-r#100` :

```
dmpmqmsg -m QM1 -i Q1 -o Q2 -r#100
```

## Déplacer les messages d'une file d'attente vers une autre file d'attente

Une variante sur [«Chargement d'une file d'attente à partir d'un fichier»](#), à la page 204. Notez la distinction entre l'utilisation de `-i` (minuscules) qui ne parcourt qu'une file d'attente et l'utilisation de `-I` (majuscules) qui extrait une file d'attente de façon destructive:

```
dmpmqmsg -m QM1 -I Q1 -o Q2
```

## Déplacer les messages de plus d'un jour d'une file d'attente vers une autre file d'attente

Cet exemple illustre l'utilisation de la sélection d'âge. Vous pouvez sélectionner des messages plus anciens, plus jeunes ou compris dans une plage d'âges.

```
dmpmqmsg -m QM1 -I Q1 -o Q2 -T1440
```

## Afficher les âges des messages actuellement dans une file d'attente

Utilisez les options suivantes sur la ligne de commande:

```
dmpmqmsg -m QM1 -i Q1 -f stdout -dT
```

## Gestion du fichier de messages

Après avoir déchargé le message de votre file d'attente, comme dans [«Déchargement d'une file d'attente dans un fichier»](#), à la page 204, vous pouvez éditer le fichier.

Vous pouvez également modifier le format du fichier afin d'utiliser l'une des options d'affichage que vous n'avez pas indiquées lors du déchargement de la file d'attente.

Vous pouvez utiliser l'utilitaire **dmpmqmsg** pour retraiter le fichier dans le format requis, même après le déchargement de la file d'attente. Utilisez les options suivantes sur la ligne de commande.

```
dmpmqmsg -f c:\oldfile -f c:\newfile -dA
```

Pour obtenir une description de la commande et de ses paramètres, voir [dmpmqmsg](#) .

## Utilisation d'objets IBM MQ distants

Vous pouvez administrer des objets IBM MQ sur des gestionnaires de files d'attente éloignées à l'aide de commandes MQSC, PCF ou administrative REST API. Avant de pouvoir utiliser l'une de ces méthodes, vous devez définir des files d'attente de transmission et des canaux entre le gestionnaire de files d'attente locales et le gestionnaire de files d'attente éloignées afin que les commandes puissent être envoyées au gestionnaire de files d'attente éloignées et les réponses reçues par le gestionnaire de files d'attente locales. Vous pouvez également configurer un cluster de gestionnaires de files d'attente, puis utiliser les mêmes méthodes d'administration à distance.

### Pourquoi et quand exécuter cette tâche

Pour préparer les gestionnaires de files d'attente pour l'administration à distance, vous devez configurer les objets suivants sur le gestionnaire de files d'attente local:

- Programme d'écoute.
- File d'attente de transmission portant le nom du gestionnaire de files d'attente éloignées.
- Canal émetteur disposant des détails de connexion pour le gestionnaire de files d'attente éloignées.
- Canal récepteur portant le même nom que le canal émetteur sur le gestionnaire de files d'attente éloignées.

Vous devez également configurer les objets suivants sur le gestionnaire de files d'attente éloignées:

- Programme d'écoute.
- File d'attente de transmission portant le nom du gestionnaire de files d'attente local.
- Canal émetteur disposant des détails de connexion pour le gestionnaire de files d'attente local.
- Canal récepteur portant le même nom que le canal émetteur sur le gestionnaire de files d'attente local.

Pour plus d'informations sur la configuration de ces objets, voir [«Configuration des gestionnaires de files d'attente pour l'administration à distance»](#), à la page 207.

Vous pouvez également configurer un cluster de gestionnaires de files d'attente. Un *cluster* est un groupe de gestionnaires de files d'attente configurés de telle sorte que les gestionnaires de files d'attente puissent communiquer directement entre eux sur un réseau unique sans file d'attente de transmission, canal et définitions de file d'attente complexes. Les clusters peuvent être configurés facilement et contiennent généralement des gestionnaires de files d'attente qui sont logiquement liés d'une manière ou d'une autre et qui doivent partager des données ou des applications. Même le plus petit cluster réduit les coûts d'administration du système.

L'établissement d'un réseau de gestionnaires de files d'attente dans un cluster implique moins de définitions que l'établissement d'un environnement de mise en file d'attente réparti traditionnel. Avec moins de définitions à créer, vous pouvez configurer ou modifier votre réseau plus rapidement et plus facilement, et réduire le risque d'erreur dans vos définitions.

Pour configurer un cluster, vous avez besoin d'une définition d'émetteur de cluster (CLUSDR) et d'un récepteur de cluster (CLUSRCVR) pour chaque gestionnaire de files d'attente. Vous n'avez pas besoin de définitions de file d'attente de transmission ni de définitions de file d'attente éloignée. Les principes de l'administration à distance sont les mêmes lorsqu'ils sont utilisés dans un cluster, mais les définitions elles-mêmes sont grandement simplifiées.

Pour plus d'informations sur la configuration d'un cluster, voir [Configuration d'un cluster de gestionnaires de files d'attente](#).

## Procédure

- Pour plus d'informations sur l'administration des objets IBM MQ distants, voir les sous-rubriques suivantes:
  - [«Configuration des gestionnaires de files d'attente pour l'administration à distance»](#), à la page 207
  - [«Gestion du serveur de commandes pour l'administration à distance»](#), à la page 211
  - [«Emission de commandes MQSC sur un gestionnaire de files d'attente éloignées»](#), à la page 212
  - [«Conversion de données entre des jeux de caractères codés»](#), à la page 214

## Configuration des gestionnaires de files d'attente pour l'administration à distance

Vous pouvez administrer un gestionnaire de files d'attente éloignées à partir d'un gestionnaire de files d'attente local à l'aide des commandes administrative REST API, MQSC ou PCF. Le gestionnaire de files d'attente éloignées peut se trouver sur le même système, dans une installation différente ou sur un système différent avec le même environnement ou un environnement IBM MQ différent. Avant de pouvoir administrer à distance un gestionnaire de files d'attente à partir d'un gestionnaire de files d'attente local, vous devez créer un canal émetteur et récepteur, un programme d'écoute et une file d'attente de transmission sur chaque gestionnaire de files d'attente. Ces canaux et files d'attente permettent d'envoyer les commandes au gestionnaire de files d'attente éloignées et de recevoir les réponses sur le gestionnaire de files d'attente local. La procédure de création de ces files d'attente et canaux est la même que vous souhaitez utiliser les commandes administrative REST API, MQSC ou PCF.

### Avant de commencer

- La procédure suivante utilise des exemples de gestionnaires de files d'attente `source.queue.manager` et `target.queue.manager`. Vous devez créer et démarrer ces gestionnaires de files d'attente sur votre système pour suivre ces étapes ou remplacer vos propres noms de gestionnaire de files d'attente dans les étapes appropriées.
- La procédure suivante utilise TCP/IP comme type de transport. Vous devez connaître l'adresse IP des deux systèmes pour effectuer cette tâche.
- La procédure suivante crée des programmes d'écoute qui utilisent les ports réseau 1818 sur le système local et 1819 sur le système distant. Vous pouvez utiliser d'autres ports, mais vous devez remplacer vos valeurs de port dans les étapes appropriées.
- Vous devez exécuter les commandes dans la procédure en local ou via une fonction réseau telle que Telnet.

## Pourquoi et quand exécuter cette tâche

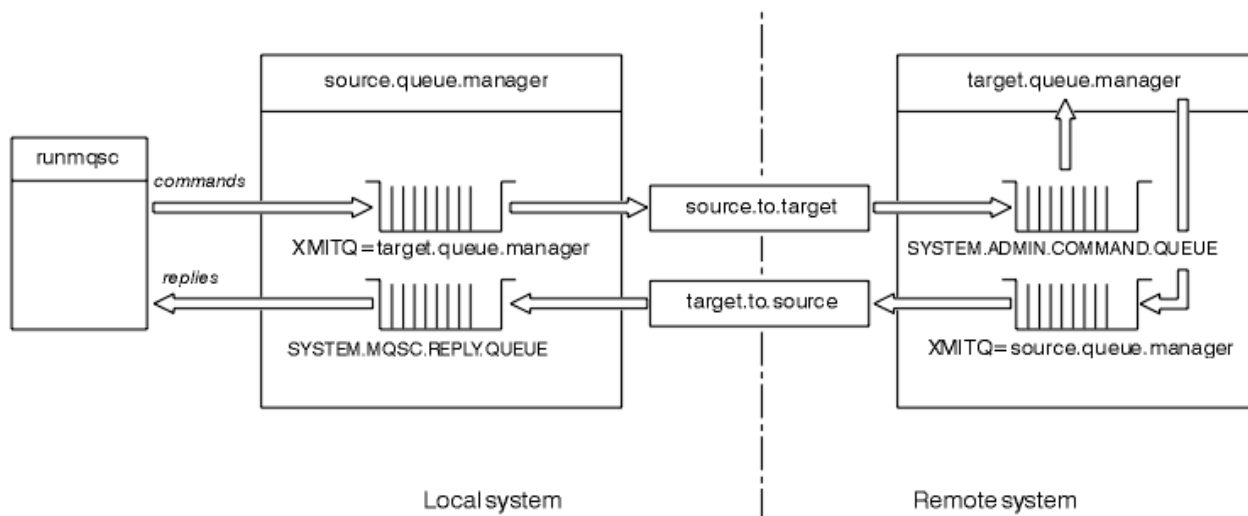


Figure 15. Configuration de canaux et de files d'attente pour l'administration à distance

La Figure 15, à la page 208 présente la configuration des gestionnaires de files d'attente, des files d'attente et des canaux dont vous avez besoin pour l'administration à distance:

- L'objet `source.queue.manager` est le gestionnaire de files d'attente source à partir duquel vous pouvez émettre des commandes administratives REST API, MQSC ou PCF et auquel les résultats de ces commandes sont renvoyés.
- L'objet `target.queue.manager` est le nom du gestionnaire de files d'attente cible, qui traite les commandes et génère des messages d'opérateur.
- Les commandes sont placées dans la file d'attente de transmission portant le même nom que le gestionnaire de files d'attente éloignées. Dans ce cas, `target.queue.manager`. Une file d'attente de transmission est une file d'attente locale spécialisée qui contient temporairement des messages avant que l'agent MCA ne les récupère et les envoie au gestionnaire de files d'attente éloignées.
- Les commandes sont envoyées par le canal `source.to.target` à `SYSTEM.ADMIN.COMMAND.QUEUE` sur le gestionnaire de files d'attente éloignées. Chaque extrémité du canal possède une définition distincte. Une extrémité est un émetteur et l'autre extrémité est un récepteur. Les deux définitions doivent avoir le même nom et constituer ensemble un canal de transmission de messages unique.
- La sortie de la commande est placée dans la file d'attente de transmission éloignée portant le même nom que le gestionnaire de files d'attente local à partir duquel la commande a été envoyée. Dans ce cas, `source.queue.manager`.
- La sortie est envoyée par le canal `target.to.source` à une file d'attente de réponses appropriée, où elle est prise et sortie par la commande d'origine.

## Procédure

1. Sur le gestionnaire de files d'attente du système distant, vérifiez que la file d'attente de commandes `SYSTEM.ADMIN.COMMAND.QUEUE` est présente. Cette file d'attente est créée par défaut lorsqu'un gestionnaire de files d'attente est créé.
2. Sur le système distant, vérifiez que le serveur de commandes est en cours d'exécution sur le gestionnaire de files d'attente. Si le serveur de commandes n'est pas en cours d'exécution, l'administration à distance n'est pas possible:
  - a) Démarrez `runmqsc` pour le gestionnaire de files d'attente. Par exemple, pour le gestionnaire de files d'attente `target.queue.manager`, entrez la commande suivante:

```
runmqsc target.queue.manager
```

- b) Affichez l'état du serveur de commandes en entrant la commande suivante:



```
DISPLAY QMSTATUS CMDSERV
```

- c) Quittez l'invite de commande **runmqsc** en entrant la commande suivante:

```
end
```

- d) Si le serveur de commandes n'est pas démarré, démarrez-le. Par exemple, pour le gestionnaire de files d'attente `target.queue.manager`, entrez la commande suivante:

```
strmqcsv target.queue.manager
```

3. Définissez les canaux, le programme d'écoute et la file d'attente de transmission sur le gestionnaire de files d'attente local:

- a) Démarrez **runmqsc** pour le gestionnaire de files d'attente. Par exemple, pour le gestionnaire de files d'attente `source.queue.manager`, entrez la commande suivante:

```
runmqsc source.queue.manager
```

- b) Définissez le canal émetteur. Ce canal émetteur doit avoir le même nom que le canal récepteur sur le gestionnaire de files d'attente éloignées. Par exemple, entrez la commande MQSC suivante, en remplaçant la valeur de **CONNNAME** par l'adresse IP du gestionnaire de files d'attente éloignées et le numéro de port du programme d'écoute:

```
DEFINE CHANNEL ('source.to.target') +  
CHLTYPE(SDR) +  
CONNNAME (localhost:1819) +  
XMITQ ('target.queue.manager') +  
TRPTYPE(TCP)
```

- c) Définissez le canal récepteur. Ce canal récepteur doit avoir le même nom que le canal émetteur sur le gestionnaire de files d'attente éloignées. Par exemple, entrez la commande suivante :

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

- d) Définissez le programme d'écoute sur le gestionnaire de files d'attente local. Par exemple, entrez la commande suivante :

```
DEFINE LISTENER ('source.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1818)
```

- e) Définissez la file d'attente de transmission sur le gestionnaire de files d'attente local. Cette file d'attente de transmission doit avoir le même nom que le gestionnaire de files d'attente éloignées. Par exemple, entrez la commande suivante :

```
DEFINE QLOCAL ('target.queue.manager') +  
USAGE (XMITQ)
```

- f) Démarrez le programme d'écoute. Par exemple, entrez la commande suivante :

```
START LISTENER ('source.queue.manager')
```

- g) Quittez l'invite de commande **runmqsc** en entrant la commande suivante:

```
end
```

4. Définissez les canaux, le programme d'écoute et la file d'attente de transmission sur le gestionnaire de files d'attente éloignées:

- a) Démarrez **runmqsc** pour le gestionnaire de files d'attente. Par exemple, pour le gestionnaire de files d'attente `target.queue.manager`, entrez la commande suivante:

```
runmqsc target.queue.manager
```

- b) Définissez le canal émetteur. Ce canal émetteur doit avoir le même nom que le canal récepteur sur le gestionnaire de files d'attente local. Par exemple, entrez la commande MQSC suivante, en remplaçant la valeur de **CONNNAME** par l'adresse IP du gestionnaire de files d'attente local et le numéro de port du programme d'écoute:

```
DEFINE CHANNEL ('target.to.source') +  
CHLTYPE(SDR) +  
CONNNAME (localhost:1818) +  
XMITQ ('source.queue.manager') +  
TRPTYPE(TCP)
```

- c) Définissez le canal récepteur. Ce canal récepteur doit avoir le même nom que le canal émetteur sur le gestionnaire de files d'attente local. Par exemple, entrez la commande suivante:

```
DEFINE CHANNEL ('source.to.target') +  
CHLTYPE(RCVR) +  
TRPTYPE(TCP)
```

- d) Définissez le programme d'écoute. Par exemple, entrez la commande suivante :

```
DEFINE LISTENER ('target.queue.manager') +  
TRPTYPE (TCP) +  
PORT (1819)
```

- e) Définissez la file d'attente de transmission. Cette file d'attente de transmission doit avoir le même nom que le gestionnaire de files d'attente local. Par exemple, entrez la commande suivante :

```
DEFINE QLOCAL ('source.queue.manager') +  
USAGE (XMITQ)
```

- f) Démarrez le programme d'écoute. Par exemple, entrez la commande suivante :

```
START LISTENER ('target.queue.manager')
```

- g) Quittez **runmqsc** en entrant la commande suivante:

```
end
```

## 5. Démarrez le canal émetteur sur le système local:

- a) Démarrez **runmqsc** pour le gestionnaire de files d'attente. Par exemple, pour le gestionnaire de files d'attente `source.queue.manager`, entrez la commande suivante:

```
runmqsc source.queue.manager
```

- b) Démarrez le canal émetteur. Par exemple, entrez la commande suivante :

```
START CHANNEL ('source.to.target')
```

- c) Quittez **runmqsc** en entrant la commande suivante:

```
end
```

## 6. Démarrez le canal émetteur sur le système distant:

- a) Démarrez **runmqsc** pour le gestionnaire de files d'attente. Par exemple, pour le gestionnaire de files d'attente `target.queue.manager`, entrez la commande suivante:

```
runmqsc target.queue.manager
```

- b) Démarrez le canal émetteur. Par exemple, entrez la commande suivante :

```
START CHANNEL ('target.to.source')
```

- c) Quittez **runmqsc** en entrant la commande suivante:

```
end
```

7. Vérifiez que la configuration a abouti en envoyant une commande MQSC depuis le système local vers le gestionnaire de files d'attente éloignées:

- a) Lancez l'invite de commande **runmqsc** pour le gestionnaire de files d'attente éloignées à partir du système local. Par exemple, entrez la commande suivante :

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- b) Affichez les files d'attente sur le gestionnaire de files d'attente éloignées en entrant la commande suivante:

```
DISPLAY QUEUE (*)
```

En cas de réussite, la liste des files d'attente du gestionnaire de files d'attente éloignées s'affiche.

- c) Si ces étapes ne fonctionnent pas, vérifiez que les canaux des deux systèmes sont en cours d'exécution. Si les canaux ne sont pas en cours d'exécution et ne démarrent pas, vérifiez que les canaux et les files d'attente de transmission sont configurés correctement et que le serveur de commandes est en cours d'exécution. Par exemple, vérifiez que le nom CONNAME correct est spécifié pour les canaux émetteurs et que les noms des files d'attente de transmission sont corrects. Recherchez également dans les journaux du gestionnaire de files d'attente des exceptions de sécurité susceptibles de vous aider à résoudre le problème.

## Résultats

Vos gestionnaires de files d'attente sont configurés pour administrer à distance le gestionnaire de files d'attente éloignées à partir du système local.

## Que faire ensuite

- En savoir plus sur l'administration à distance à l'aide des commandes MQSC: [«Emission de commandes MQSC sur un gestionnaire de files d'attente éloignées»](#), à la page 212
- En savoir plus sur l'écriture de programmes d'administration à l'aide des commandes PCF: [«Utilisation des formats de commande programmables IBM MQ»](#), à la page 28.
- En savoir plus sur l'utilisation de l'administrative REST API pour l'administration à distance: [«Administration à distance à l'aide de REST API»](#), à la page 82.

## Gestion du serveur de commandes pour l'administration à distance

Chaque gestionnaire de files d'attente est associé à un serveur de commandes. Un serveur de commandes traite toutes les commandes entrantes des gestionnaires de files d'attente éloignées ou les commandes PCF des applications. Il présente les commandes au gestionnaire de files d'attente pour traitement et renvoie un code achèvement ou un message d'opérateur. Vous pouvez démarrer, arrêter et afficher le statut du serveur de commandes. Un serveur de commandes est obligatoire pour toute administration impliquant des commandes PCF, MQAI, ainsi que pour l'administration à distance.

## Avant de commencer

En fonction de la valeur de l'attribut de gestionnaire de files d'attente, **SCMDSERV**, le serveur de commandes est démarré automatiquement au démarrage du gestionnaire de files d'attente ou doit être

démarré manuellement. Si le serveur de commandes est démarré automatiquement, vous ne pouvez pas utiliser les commandes `strmqcsv` ou `endmqcsv` pour démarrer et arrêter le serveur de commandes. Vous pouvez modifier la valeur de l'attribut **SCMDSERV** à l'aide de la commande MQSC **ALTER QMGR**. Par défaut, le serveur de commandes est démarré automatiquement.

L'arrêt d'un gestionnaire de files d'attente arrête également le serveur de commandes qui lui est associé.

## Procédure

- Affichez l'état du serveur de commandes:
  - a) Démarrez l'invite de commande **runmqsc** pour le gestionnaire de files d'attente approprié en entrant la commande suivante:

```
runmqsc target.queue.manager
```

où `target.queue.manager` est le gestionnaire de files d'attente pour lequel le serveur de commandes est affiché.

- b) Affichez le statut du serveur de commandes en entrant la commande MQSC suivante:

```
DISPLAY QMSTATUS CMDSERV
```

- c) Quittez l'invite de commande **runmqsc** en entrant la commande suivante:

```
end
```

- Si le serveur de commandes n'est pas configuré pour démarrer automatiquement, démarrez-le en entrant la commande suivante:

```
strmqcsv target.queue.manager
```

où `target.queue.manager` est le gestionnaire de files d'attente pour lequel le serveur de commandes est démarré.

- Si le serveur de commandes n'est pas configuré pour démarrer automatiquement, arrêtez-le en entrant la commande suivante:

```
endmqcsv target.queue.manager
```

où `target.queue.manager` est le gestionnaire de files d'attente pour lequel le serveur de commandes est arrêté.

Par défaut, le serveur de commandes s'arrête de manière contrôlée. Vous pouvez arrêter le serveur de commandes immédiatement en ajoutant l'indicateur `-i` à la commande.

## Emission de commandes MQSC sur un gestionnaire de files d'attente éloignées

Après avoir configuré les gestionnaires de files d'attente pour l'administration à distance, vous pouvez utiliser une forme particulière de la commande **runmqsc** sur un système local pour exécuter des commandes MQSC sur un gestionnaire de files d'attente éloignées. Chaque commande est envoyée en tant que PCF d'échappement à la file d'attente de commandes, `SYSTEM.ADMIN.COMMAND.QUEUE`, du gestionnaire de files d'attente éloignées. Les réponses sont reçues sur le système `SYSTEM.MQSC.REPLY.QUEUE`.

### Avant de commencer

Vous devez effectuer les étapes de la rubrique «Configuration des gestionnaires de files d'attente pour l'administration à distance», à la page 207 pour configurer les canaux, les files d'attente de transmission,

les programmes d'écoute et le serveur de commandes avant de pouvoir administrer à distance un gestionnaire de files d'attente à l'aide des commandes MQSC.

## Procédure

1. Vérifiez que le serveur de commandes est en cours d'exécution sur le gestionnaire de files d'attente éloignées.

Pour plus d'informations sur le démarrage du serveur de commandes sur un gestionnaire de files d'attente, voir [«Gestion du serveur de commandes pour l'administration à distance»](#), à la page 211.

2. Sur le gestionnaire de files d'attente source, vous pouvez ensuite exécuter les commandes MQSC de l'une des deux manières suivantes:

- En mode interactif, en démarrant **runmqsc** avec les commandes suivantes:

- Si le gestionnaire de files d'attente éloignées se trouve sous z/OS, entrez la commande suivante:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager
```

- Si le gestionnaire de files d'attente éloignées se trouve sur Multiplatforms, entrez la commande suivante:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager
```

- A partir d'un fichier de commandes:

- a. Placez les commandes MQSC à exécuter sur le système distant dans un fichier texte, une commande par ligne.

- b. Vérifiez vos commandes MQSC sur le gestionnaire de files d'attente local à l'aide de l'indicateur **-v** de la commande **runmqsc**. L'indicateur **-v** vérifie que les commandes sont valides, mais ne les exécute pas. Sachez que certaines commandes peuvent échouer si elles sont applicables au gestionnaire de files d'attente éloignées mais non au gestionnaire de files d'attente local:

```
runmqsc -v source.queue.manager < myCmdFile.in > results.out
```

`myCmdFile.in` contient les commandes MQSC à vérifier et le fichier `results.out` contient les résultats de vérification des commandes.

- c. Exécutez le fichier de commandes sur le gestionnaire de files d'attente éloignées en entrant l'une des commandes suivantes:

- Si le gestionnaire de files d'attente éloignées se trouve sous z/OS, entrez la commande suivante:

```
runmqsc -w 30 -x -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

- Si le gestionnaire de files d'attente éloignées se trouve sur Multiplatforms, entrez la commande suivante:

```
runmqsc -w 30 -m source.queue.manager target.queue.manager < myCmdFile.in >  
results.out
```

Les paramètres utilisés sont les suivants:

### **-w secondes**

Indique que les commandes MQSC sont exécutées en mode indirect, où les commandes sont placées dans la file d'attente d'entrée du serveur de commandes et exécutées dans l'ordre.

La variable *secondes* indique la durée d'attente, en secondes, d'une réponse du gestionnaire de files d'attente éloignées. Toutes les réponses reçues après cette période sont supprimées, mais les commandes MQSC continuent de s'exécuter sur le gestionnaire de files d'attente éloignées. Le

message suivant est généré sur le gestionnaire de files d'attente local lorsque la commande arrive à expiration:

```
AMQ8416: MQSC timed out waiting for a response from the command server.
```

Lorsque vous arrêtez d'émettre des commandes MQSC, le gestionnaire de files d'attente local affiche toutes les réponses arrivées à expiration et supprime toutes les autres réponses.

**-x**

Indique que le gestionnaire de files d'attente éloignées est un gestionnaire de files d'attente z/OS .

**-m *NomGestFilesLocal***

Indique le nom du gestionnaire de files d'attente local que vous souhaitez utiliser pour soumettre des commandes au gestionnaire de files d'attente éloignées.

## Que faire ensuite

Si vous avez des difficultés à exécuter des commandes MQSC à distance:

- Vérifiez que le gestionnaire de files d'attente éloignées est en cours d'exécution.
- Vérifiez que le serveur de commandes est en cours d'exécution sur le système distant.
- Vérifiez que l'intervalle de déconnexion du canal n'a pas expiré. Par exemple, si un canal a démarré, puis s'est arrêté après un certain temps. Ceci est particulièrement important si vous démarrez les canaux manuellement.
- Assurez-vous que les demandes envoyées à partir du gestionnaire de files d'attente local sont bien ciblées sur le gestionnaire de files d'attente cible. Par exemple, les demandes qui incluent des paramètres qui ne sont pas pris en charge sur le gestionnaire de files d'attente éloignées.
- Voir aussi [Résolution des problèmes avec les commandes MQSC](#).

## Conversion de données entre des jeux de caractères codés

Les données de message dans des formats définis par IBM MQ (également appelés formats intégrés) peuvent être converties par le gestionnaire de files d'attente d'un jeu de caractères codés à un autre, à condition que les deux jeux de caractères soient associés à une langue unique ou à un groupe de langues similaires.

Par exemple, la conversion entre les jeux de caractères codés avec les identificateurs (CCSID) 850 et 500 est prise en charge, car les deux s'appliquent aux langues d'Europe occidentale.

Pour les conversions de caractères de nouvelle ligne (NL) EBCDIC en ASCII, voir [Toutes les sections des gestionnaires de files d'attente du fichier mqs.ini](#) et la variable d'environnement **AMQ\_CONVEBCDICNEWLINE** .

Les conversions prises en charge sont définies dans [Traitement de la conversion de données](#).

La conversion entre les CCSID 37 et 500 est prise en charge sur les systèmes IBM MQ Appliance, Windows, Linux et macOS.

### Lorsqu'un gestionnaire de files d'attente ne parvient pas à convertir des messages dans des formats intégrés

Le gestionnaire de files d'attente ne peut pas convertir automatiquement les messages dans des formats intégrés si leurs CCSID représentent des groupes de langues nationales différents. Par exemple, la conversion entre le CCSID 850 et le CCSID 1025 (qui est un jeu de caractères codés EBCDIC pour les langues utilisant le script cyrillique) n'est pas prise en charge car la plupart des caractères d'un jeu de caractères codés ne peuvent pas être représentés dans l'autre. Si vous disposez d'un réseau de gestionnaires de files d'attente travaillant dans des langues nationales différentes et que la conversion de données entre certains jeux de caractères codés n'est pas prise en charge, vous pouvez activer une conversion par défaut.

Pour les plateformes auxquelles le `ccsid_part2.tbl` s'applique, voir [«Spécification de la conversion de données par défaut»](#), à la page 218 using `ccsid_part2.tbl` pour plus d'informations. La conversion de données par défaut sur les plateformes autres que celles auxquelles s'applique le fichier `ccsid_part2.tbl` est décrite dans [«Conversion de données par défaut»](#), à la page 216.

## Support de la conversion de données Unicode amélioré

Le produit prend en charge tous les caractères Unicode définis dans la norme Unicode 8.0 dans la conversion de données. Cela inclut la prise en charge complète de UTF-16, y compris les paires de substitution (une paire de caractères UTF-16 de 2 octets dans la plage X'D800' à X'DFFF' qui représentent un point de code Unicode au-dessus de U+FFFF).

La combinaison de séquences de caractères est également prise en charge dans les cas où un caractère précomposé dans un CCSID est mappé à une séquence de caractères de combinaison dans un autre CCSID.

La conversion de données vers et depuis Unicode et les CCSID 1388, 1390, 1399, 4933, 5488 et 16884 a été étendue, sur certaines plateformes, pour prendre en charge tous les points de code actuellement définis pour ces CCSID, y compris ceux qui correspondent à des points de code dans des plans supplémentaires Unicode.

Dans le cas des CCSID 1390, 1399 et 16884, cela inclut les caractères définis dans la norme JIS X 0213 (JIS2004).

La prise en charge a également été ajoutée pour la conversion vers et depuis Unicode et six nouveaux CCSID (1374 à 1379).



## Fichier `ccsid_part2.tbl`


Un fichier supplémentaire, `ccsid_part2.tbl`, est fourni.


Le fichier `ccsid_part2.tbl` a priorité sur le fichier `ccsid.tbl` et :

- D'ajouter ou de modifier des entrées de CCSID
- De spécifier une conversion de données par défaut
- De spécifier des données pour différents niveaux de commande

`ccsid_part2.tbl` s'applique uniquement aux plateformes suivantes :

-  Linux - toutes les versions
-  Windows

 Sous IBM MQ for Windows, `ccsid_part2.tbl` se trouve dans le répertoire `MQDataRoot\conv\table` par défaut. En outre, sous IBM MQ for Windows, il enregistre tous les jeux de codes pris en charge.

 Sous IBM MQ for Linux, `ccsid_part2.tbl` se trouve dans le répertoire `MQDataRoot/conv/table` et les jeux de codes pris en charge se trouvent dans les tables de conversion fournies par IBM MQ.

Bien que le fichier `ccsid_part2.tbl` remplace le fichier `ccsid.tbl` existant utilisé dans les versions précédentes de IBM MQ pour fournir des informations CCSID supplémentaires, le fichier `ccsid.tbl` continue d'être analysé par IBM MQ et ne doit donc pas être supprimé.

Pour plus d'informations, voir [«Le fichier `ccsid\_part2.tbl`»](#), à la page 217.

## Fichier `ccsid.tbl`

Sur les plateformes autres que celles auxquelles `ccsid_part2.tbl` s'applique, le fichier `ccsid.tbl` est utilisé aux fins suivantes:

- **AIX** Sous AIX, les jeux de codes pris en charge sont détenus en interne par le système d'exploitation.
- Il indique tous les jeux de codes supplémentaires. Pour spécifier des jeux de codes supplémentaires, vous devez éditer `ccsid.tbl` (des conseils sur la procédure à suivre sont fournis dans le fichier).
- Il indique toute conversion de données par défaut.

Vous pouvez mettre à jour les informations enregistrées dans `ccsid.tbl`. Vous pouvez être amené à le faire si, par exemple, une version ultérieure de votre système d'exploitation prend en charge des jeux de caractères codés supplémentaires.

## Conversion de données par défaut

Si vous configurez des canaux entre deux machines sur lesquelles la conversion de données n'est pas normalement prise en charge, vous devez activer la conversion de données par défaut pour que les canaux fonctionnent.

Sur les plateformes autres que celles auxquelles `ccsid_part2.tbl` s'applique, pour activer la conversion de données par défaut, éditez le fichier `ccsid.tbl` pour spécifier un CCSID EBCDIC par défaut et un CCSID ASCII par défaut. Des instructions sur la façon de procéder sont incluses dans le fichier. Vous devez effectuer cette opération sur toutes les machines qui seront connectées à l'aide des canaux. Redémarrez le gestionnaire de files d'attente pour que la modification soit prise en compte.

Le processus de conversion de données par défaut est le suivant:

- Si la conversion entre les CCSID source et cible n'est pas prise en charge, mais que les CCSID des environnements source et cible sont EBCDIC ou ASCII, les données de type caractères sont transmises à l'application cible sans conversion.
- Si un CCSID représente un jeu de caractères codés ASCII et que l'autre représente un jeu de caractères codés EBCDIC, IBM MQ convertit les données à l'aide des CCSID de conversion de données par défaut définis dans `ccsid.tbl`.

**Remarque :** Essayez de limiter les caractères convertis à ceux qui ont les mêmes valeurs de code dans le jeu de caractères codés indiqué pour le message et dans le jeu de caractères codés par défaut. Si vous utilisez uniquement l'ensemble de caractères qui est valide pour les noms d'objet IBM MQ (tel que défini dans [Nommage des objets IBM MQ](#)) vous satisferez, en général, à cette exigence. Des exceptions se produisent avec les CCSID EBCDIC 290, 930, 1279 et 5026 utilisés au Japon, où les caractères minuscules ont des codes différents de ceux utilisés dans les autres CCSID EBCDIC.

## Conversion de messages dans des formats définis par l'utilisateur

Le gestionnaire de files d'attente ne peut pas convertir des messages dans des formats définis par l'utilisateur d'un jeu de caractères codés à un autre. Si vous devez convertir des données dans un format défini par l'utilisateur, vous devez fournir un exit de conversion de données pour chaque format. N'utilisez pas les CCSID par défaut pour convertir les données de type caractères dans des formats définis par l'utilisateur. Pour plus d'informations sur la conversion de données dans des formats définis par l'utilisateur et sur l'écriture d'exits de conversion de données, voir [Ecriture d'exits de conversion de données](#).

## Modification du CCSID du gestionnaire de files d'attente

Une fois que vous avez utilisé l'attribut **CCSID** de la commande **ALTER QMGR** pour modifier le CCSID du gestionnaire de files d'attente, arrêtez et redémarrez le gestionnaire de files d'attente pour vous assurer que toutes les applications en cours d'exécution, y compris le serveur de commandes et les programmes de canal, sont arrêtées et redémarrées.

Cette opération est nécessaire car toutes les applications en cours d'exécution lorsque le CCSID du gestionnaire de files d'attente est modifié continuent d'utiliser le CCSID existant.



## Le fichier `ccsid_part2.tbl`

Le fichier `ccsid_part2.tbl` est utilisé pour fournir des informations CCSID supplémentaires. Le fichier `ccsid_part2.tbl` remplace le fichier `ccsid.tbl` utilisé avant IBM MQ 9.0.

**Remarque :** Le fichier `ccsid.tbl`, qui a été utilisé avant IBM MQ 9.0 pour fournir des informations CCSID supplémentaires, continue d'être analysé par IBM MQ et ne doit pas être supprimé. Toutefois, les entrées de `ccsid_part2.tbl` sont prioritaires sur les autres entrées de `ccsid.tbl`.



Vous devez utiliser `ccsid_part2.tbl` au lieu de `ccsid.tbl` car `ccsid_part2.tbl`:

- Contient la prise en charge des valeurs de codage Unicode. Depuis la IBM MQ 9.0, le produit prend en charge tous les caractères Unicode définis dans la norme Unicode 8.0 dans la conversion de données, y compris la prise en charge complète de UTF-16. Pour plus d'informations, voir [«Conversion de données entre des jeux de caractères codés»](#), à la page 214.
- Permet d'indiquer la version des entrées CCSID de sorte qu'elles ne s'appliquent qu'aux niveaux de commande sélectionnés.



Vous pouvez utiliser le fichier `ccsid_part2.tbl` pour:

- Ajouter ou modifier des entrées CCSID
- De spécifier une conversion de données par défaut
- De spécifier des données pour différents niveaux de commande

Le fichier `ccsid_part2.tbl` concerne uniquement les plateformes suivantes :

-  Linux - toutes les versions
-  Windows

L'emplacement du fichier `ccsid_part2.tbl` dépend de votre plateforme:

-  Répertoire `MQDataRoot/conv/table` sur toutes les versions de Linux.
-  Le répertoire `MQDataRoot\conv\table` sous Windows.

## Ajout ou modification d'entrées CCSID

Une entrée du fichier `ccsid_part2.tbl` a le format suivant:

```
<CCSID number> <Base CCSID> <DBCS CodePage> <SBCS CodePage>  
<Type> <Encoding> <ACRI> <Name>
```

Voici un exemple d'entrée pour CCSID 1200 (UTF-16):

```
1200 1200 1200 1200 3 8 0 UTF-16
```

**Remarque :** Pour plus de détails sur la valeur d'ACRI, voir le commentaire dans le fichier `ccsid_part2.tbl`.

Au format `ccsid_part2.tbl` :

### Le type peut être égal à:

- 1=SBCS
- 2=DBCS
- 3=MBCS

### Le codage peut être égal à:

- 1=EBCDIC
- 2 = ASCII
- 3 = ISO

- 4 = UCS-2
- 5 = UTF-8
- 6 = Euc
- 7 = GB18030
- 8 = UTF-16
- 9 = UTF-32

Lorsque vous éditez le fichier, vous:

- Peut spécifier un commentaire en utilisant le symbole # au début d'une ligne. Cela empêche IBM MQ de tenter d'analyser la ligne.
- Impossible de fournir des commentaires en ligne.
- Vous devez vous assurer que vous ne créez pas de lignes vides.
- Vous ne devez pas ajouter de nouvelles entrées à la fin du fichier.

De nouvelles entrées CCSID doivent être ajoutées avant les informations de la table ACRI.

## Spécification de la conversion de données par défaut

Vous pouvez définir des CCSID de conversion par défaut, qui sont utilisés pour la conversion entre des CCSID ASCII ou similaires et des CCSID EBCDIC, si aucune conversion n'est prise en charge entre deux CCSID.

Si vous activez cette fonction, la conversion par défaut est utilisée pour la transmission et les en-têtes de message, et peut également être utilisée dans la conversion des données utilisateur.

Les conversions par défaut sont activées en créant deux lignes similaires aux suivantes:

default	0	500	1	1	0
default	0	850	1	2	0

La première ligne définit la valeur par défaut pour les CCSID EBCDIC sur 500 et la deuxième ligne définit la valeur par défaut pour les CCSID ASCII et similaires sur 850.

## Spécification de données pour différents niveaux de commande

Pour spécifier des entrées CCSID pour les différents niveaux de commande de IBM MQ, utilisez un symbole deux points suivi du niveau de commande (ou des niveaux de commande) de IBM MQ auquel vous souhaitez que la section suivante soit applicable.

Ce nombre représente le niveau de commande minimal dans lequel le gestionnaire de files d'attente ou le client doit s'exécuter. Par exemple, si le gestionnaire de files d'attente en cours est au niveau de commande 900 et qu'il rencontre un indicateur de niveau de commande 800 ou 900, les CCSID sont lus.

Toutefois, un gestionnaire de files d'attente de niveau 800 ignore les CCSID de la section 900.

Le niveau de commande indiqué est applicable à toutes les entrées CCSID détectées après un indicateur de niveau de commande, jusqu'à ce qu'un nouvel indicateur de niveau de commande soit trouvé.

Si vous devez définir le niveau de commande sur tous les niveaux de commande, indiquez le nombre zéro.

Lors de la première analyse syntaxique de `ccsid_part2.tbl`, IBM MQ traite tous les CCSID détectés comme valides pour tous les niveaux de commande de IBM MQ.

La gestion des versions commence à être utilisée uniquement lorsque IBM MQ rencontre le premier indicateur de niveau de commande.

Le fragment de code suivant illustre un exemple d'utilisation de la gestion des versions:

```
# Comment Block
# End of Comment Block
```

```

# Because no command level flag is specified and we're at the start of the file
# the following CCSIDs will be read on all versions
  819  819  0  819  1  3  0  IS08859-1
  923  923  0  923  1  3  0  IS08859-15
 1051 1051  0  1051 1  3  0  IBM-1051
# The colon :900 below shows that the CCSIDs after will only be for MQ cmd level 900 and above
:900
 8629 437  0  437  1  2  0  IBM-437
12725 437  0  437  1  2  0  IBM-437
16821 437  0  437  1  2  0  IBM-437
20917 437  0  437  1  2  0  IBM-437
# The colon :0 below shows that the CCSIDs after will be for all version of MQ
:0
 4946 850  0  850  1  2  0  IBM-850
33618 850  0  850  1  2  0  IBM-850
61697 850  0  850  1  2  0  IBM-850
61698 850  0  850  1  2  0  IBM-850

```

## Administration de Managed File Transfer

Utilisez les commandes Managed File Transfer pour administrer Managed File Transfer. Vous pouvez également utiliser le IBM MQ Explorer pour certaines des tâches d'administration.

### Démarrer le transfert en plaçant un message dans une file d'attente de commandes d'agent

Vous pouvez également démarrer un transfert de fichier en plaçant un message de transfert de fichier dans la file d'attente de commandes de l'agent source. Exemple de nom de file d'attente de commandes: SYSTEM.FTE.COMMAND.AGENT01. Vous devez vous assurer que le message atteint la file d'attente de commandes de l'agent source correct ; si le message est reçu par un agent qui ne correspond pas aux informations source dans le XML, le message est rejeté.

Le XML de la demande de transfert doit être conforme au schéma `FileTransfer.xsd` et utiliser l'élément `<request>` comme élément racine. Pour plus d'informations sur la structure et le contenu d'un message de demande de transfert, voir [Format de message de demande de transfert de fichier](#). La façon dont vous placez le message de demande de transfert dans une file d'attente de commandes d'agent est spécifique à la tâche. Par exemple, vous pouvez utiliser l'API IBM MQ Java pour placer un message dans la file d'attente à l'aide d'un programme.

#### Concepts associés

[«Transfert de données de fichiers vers des messages», à la page 275](#)

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ.

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

[«Le pont de protocole», à la page 302](#)

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT, dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

[«Utilisation de MFT à partir de IBM Integration Bus», à la page 342](#)

Vous pouvez utiliser Managed File Transfer depuis IBM Integration Bus à l'aide des noeuds FTEOutput et FTEInput.

[«MFT Reprise et redémarrage», à la page 343](#)

Si votre agent ou votre gestionnaire de files d'attente n'est pas disponible pour une raison quelconque, par exemple en raison d'une panne d'alimentation ou de réseau, Managed File Transfer effectue la récupération suivante dans les scénarios suivants:

### **Tâches associées**

«Démarrage d'un agent MFT», à la page 220

Avant de pouvoir utiliser un agent Managed File Transfer pour un transfert de fichier, vous devez d'abord démarrer l'agent.

«Démarrage d'un nouveau transfert de fichiers», à la page 228

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

«Surveillance des transferts de fichiers en cours», à la page 235

Vous pouvez surveiller un transfert de fichiers en cours à l'aide de l'onglet **Managed File Transfer-Current Transfer Progress** dans IBM MQ Explorer. Ce transfert de fichier peut être démarré à partir de IBM MQ Explorer ou de la ligne de commande. L'onglet affiche également la progression des transferts planifiés au point de début des transferts planifiés.

«Affichage du statut des transferts de fichiers dans le journal de transfert», à la page 237

Vous pouvez afficher les détails des transferts de fichiers à l'aide du **Journal de transfert** dans IBM MQ Explorer. Il peut s'agir de transferts démarrés à partir de la ligne de commande ou du IBM MQ Explorer. Vous pouvez également personnaliser les éléments affichés dans le **Journal de transfert**.

«Surveillance des ressources MFT», à la page 239

Vous pouvez surveiller les ressources Managed File Transfer ; par exemple, une file d'attente ou un répertoire. Lorsqu'une condition est satisfaite sur cette ressource, le moniteur de ressources démarre une tâche, telle qu'un transfert de fichier. Vous pouvez créer un moniteur de ressources à l'aide de la commande **fteCreateMonitor** ou de la vue **Moniteurs** dans le plug-in Managed File Transfer pour IBM MQ Explorer.

«Utilisation des modèles de transfert de fichiers», à la page 272

Vous pouvez utiliser des modèles de transfert de fichiers pour stocker des paramètres de transfert de fichiers communs pour des transferts répétés ou complexes. Créez un modèle de transfert à partir de la ligne de commande à l'aide de la commande **fteCreateTemplate** ou utilisez la IBM MQ Explorer pour créer un modèle de transfert à l'aide de l'assistant **Créer un modèle pour Managed File Transfer**, ou sauvegardez un modèle lors de la création d'un transfert de fichier en cochant la case **Sauvegarder les paramètres de transfert en tant que modèle**. La fenêtre **Modèles de transfert** affiche tous les modèles de transfert que vous avez créés dans votre réseau Managed File Transfer.

«Liste des agents MFT», à la page 226

Vous pouvez répertorier les agents Managed File Transfer enregistrés auprès d'un gestionnaire de files d'attente particulier à l'aide de la ligne de commande ou de l' IBM MQ Explorer.

«Arrêt d'un agent MFT», à la page 226

Vous pouvez arrêter un agent Managed File Transfer à partir de la ligne de commande. Lorsque vous arrêtez un agent, vous mettez l'agent au repos et vous l'autorisez à terminer son transfert de fichiers en cours avant de l'arrêter. Vous pouvez également spécifier le paramètre **-i** sur la ligne de commande pour arrêter immédiatement un agent. Une fois l'agent arrêté, vous ne pouvez pas l'utiliser pour transférer des fichiers tant que vous ne l'avez pas redémarré.

Configuration d'un consignateur MFT

### **Référence associée**

Instructions pour le transfert de fichiers

## **Démarrage d'un agent MFT**

Avant de pouvoir utiliser un agent Managed File Transfer pour un transfert de fichier, vous devez d'abord démarrer l'agent.

## Pourquoi et quand exécuter cette tâche

Vous pouvez démarrer un Managed File Transfer Agent à partir de la ligne de commande. Dans ce cas, le processus de l'agent s'arrête lorsque vous vous déconnectez du système.

**ALW** Sous AIX, Linux, and Windows, vous pouvez configurer un agent pour qu'il continue de s'exécuter lorsque vous vous déconnectez du système et qu'il puisse continuer à recevoir des transferts de fichiers.

**z/OS** Sous z/OS, vous pouvez configurer l'agent pour qu'il démarre en tant que tâche démarrée à partir de JCL sans qu'une session interactive soit nécessaire.

Notez que si un agent rencontre une erreur irrémédiable lors de son exécution, une capture de données de première défaillance (FDC) est générée et l'agent est arrêté.

## Procédure

- Pour démarrer un agent à partir de la ligne de commande, utilisez la commande **fteStartAgent** .  
Pour plus d'informations, voir [fteStartAgent](#).
- **ALW**  
Pour configurer un agent afin qu'il continue de s'exécuter lorsque vous vous déconnectez du système:
  - **Windows** Sous Windows, configurez l'agent pour qu'il s'exécute en tant que service Windows .  
Pour plus d'informations, voir [«Démarrage d'un agent MFT en tant que service Windows»](#), à la page 221.
  - **Linux** **AIX** Sous AIX and Linux, configurez l'agent pour qu'il démarre automatiquement lors d'un réamorçage à l'aide d'un fichier script. Pour plus d'informations, voir [«Démarrage d'un agent MFT au démarrage du système AIX and Linux»](#), à la page 223.
- **z/OS**  
Sous z/OS, configurez l'agent pour qu'il démarre en tant que tâche démarrée à partir de JCL sans qu'une session interactive soit nécessaire.  
Pour plus d'informations, voir [«Starting an MFT agent on z/OS»](#), à la page 225.

### Tâches associées

[«Liste des agents MFT»](#), à la page 226

Vous pouvez répertorier les agents Managed File Transfer enregistrés auprès d'un gestionnaire de files d'attente particulier à l'aide de la ligne de commande ou de l' IBM MQ Explorer.

[«Arrêt d'un agent MFT»](#), à la page 226

Vous pouvez arrêter un agent Managed File Transfer à partir de la ligne de commande. Lorsque vous arrêtez un agent, vous mettez l'agent au repos et vous l'autorisez à terminer son transfert de fichiers en cours avant de l'arrêter. Vous pouvez également spécifier le paramètre **-i** sur la ligne de commande pour arrêter immédiatement un agent. Une fois l'agent arrêté, vous ne pouvez pas l'utiliser pour transférer des fichiers tant que vous ne l'avez pas redémarré.

### Référence associée

[Valeurs de statut de l'agent MFT](#)

### [fteStartAgent](#)

## **Windows** Démarrage d'un agent MFT en tant que service Windows

Vous pouvez démarrer un agent en tant que service Windows de sorte que lorsque vous vous déconnectez de Windows, votre agent continue de s'exécuter et puisse recevoir des transferts de fichiers.

## Pourquoi et quand exécuter cette tâche

Sous Windows, lorsque vous démarrez un agent à partir de la ligne de commande, le processus de l'agent s'exécute à l'aide du nom d'utilisateur que vous avez utilisé pour vous connecter à Windows. Lorsque vous vous déconnectez du système, le processus de l'agent s'arrête. Pour empêcher l'arrêt de l'agent, vous pouvez configurer un agent pour qu'il s'exécute en tant que service Windows. L'exécution en tant que service Windows vous permet également de configurer les agents pour qu'ils soient démarrés automatiquement lorsque l'environnement Windows démarre ou est redémarré.

Procédez comme suit pour démarrer un agent qui s'exécute en tant que service Windows. Vous devez exécuter Managed File Transfer sur l'une des versions Windows prises en charge pour exécuter l'agent en tant que service Windows. Pour la liste des environnements pris en charge, voir [System Requirements for IBM MQ](#).

Les étapes exactes varient selon que vous avez déjà créé un agent ou que vous créez un agent. Les deux options sont décrites dans les étapes suivantes.

## Procédure

1. Si vous créez un agent Managed File Transfer, utilisez la commande **fteCreateAgent**, **fteCreateCDAgent** ou **fteCreateBridgeAgent**. Indiquez le paramètre **-s** pour exécuter l'agent en tant que service Windows. Dans l'exemple suivant, l'agent AGENT1 est créé avec un gestionnaire de files d'attente d'agent QMGR1. Le service Windows s'exécute à l'aide du nom d'utilisateur fteuser, qui est associé à un mot de passe ftepassword.

```
fteCreateAgent -agentName AGENT1 -agentQMGR1 QMGR1 -s -su fteuser -sp ftepassword
```

Vous pouvez éventuellement spécifier un nom pour le service après le paramètre **-s**. Si vous n'indiquez pas de nom, le service est nommé mqmftAgentAGENTQMGR, où AGENT est le nom d'agent que vous avez indiqué et QMGR est le nom de votre gestionnaire de files d'attente d'agent. Dans cet exemple, le nom par défaut du service est mqmftAgentAGENT1QMGR1.

**Remarque :** Le compte utilisateur Windows que vous spécifiez à l'aide du paramètre **-su** doit disposer des droits **Log on as a service**. Pour plus d'informations sur la configuration, voir [Traitement des incidents liés à un agent ou à un consignateur MFT s'exécutant en tant que service Windows](#).

Pour plus d'informations, voir [fteCreateAgent](#), [fteCreateCDAgent: create a Connect:Direct bridge agent](#) ou [fteCreateBridgeAgent](#) (création et configuration d'un agent de pont de protocole MFT).

2. Si vous avez suivi l'étape précédente pour créer un agent, exécutez les commandes MQSC générées par la commande **fteCreateAgent**, **fteCreateCDAgent** ou **fteCreateBridgeAgent**. Ces commandes créent les files d'attente IBM MQ requises par l'agent.  
Par exemple, pour un agent nommé AGENT1, un gestionnaire de files d'attente d'agent nommé QMGR1 et un gestionnaire de files d'attente de coordination nommé COORDQMGR1, exécutez la commande suivante:

```
runmqsc QMGR1 MQ_DATA_PATH\mqft\config\COORDQMGR1\agents\AGENT1\AGENT1_create.mqsc
```

3. Si vous n'avez pas suivi les étapes précédentes pour créer un agent et que vous souhaitez à la place configurer un agent existant pour qu'il s'exécute en tant que service Windows, arrêtez d'abord votre agent s'il est en cours d'exécution, puis modifiez sa configuration.
  - a) L'exemple suivant utilise un agent nommé AGENT1. Exécutez ensuite la commande suivante :

```
fteStopAgent AGENT1
```

- b) Utilisez la commande **fteModifyAgent** pour configurer l'agent pour qu'il s'exécute en tant que service Windows :

```
fteModifyAgent -agentName AGENT1 -s -su fteuser -sp ftepassword
```

Pour plus d'informations, voir [fteModifyAgent: exécutez un agent MFT en tant que service Windows](#).

4. Démarrez votre agent à l'aide de la commande **fteStartAgent** . Vous pouvez également utiliser l'outil Windows Services, disponible à partir des outils d'administration du panneau de configuration, sélectionné dans le menu Démarrer du bureau Windows , pour démarrer le service.

```
fteStartAgent AGENT1
```

Le service continue de s'exécuter même si vous vous déconnectez de Windows. Pour vous assurer que le service redémarre également lorsque Windows redémarre après un arrêt, la zone **Type de démarrage** de l'outil Windows Services est définie sur **Automatique** par défaut. Remplacez cette valeur par **Manuel** si vous ne souhaitez pas que le service redémarre lorsque Windows redémarre.

5. Facultatif : Pour arrêter l'agent, utilisez la commande [fteStopAgent](#) ou l'outil Windows Services. Par exemple, à partir de la ligne de commande, exécutez la commande suivante:

```
fteStopAgent AGENT1
```

- Lorsque vous exécutez la commande **fteStopAgent** en tant que service, la commande s'exécute toujours à l'aide du paramètre **-i** , que vous ayez ou non spécifié ce paramètre. Le paramètre **-i** arrête immédiatement l'agent sans effectuer les transferts en cours. Cela est dû à une limitation du service Windows .

## Que faire ensuite

Si vous rencontrez des problèmes lors du démarrage de votre service Windows , voir [Traitement des incidents liés à un agent ou à un consignateur MFT s'exécutant en tant que service Windows](#). Cette rubrique décrit également l'emplacement des fichiers journaux du service Windows .

### Tâches associées

«Liste des agents MFT», à la page 226

Vous pouvez répertorier les agents Managed File Transfer enregistrés auprès d'un gestionnaire de files d'attente particulier à l'aide de la ligne de commande ou de l' IBM MQ Explorer.

«Arrêt d'un agent MFT», à la page 226

Vous pouvez arrêter un agent Managed File Transfer à partir de la ligne de commande. Lorsque vous arrêtez un agent, vous mettez l'agent au repos et vous l'autorisez à terminer son transfert de fichiers en cours avant de l'arrêter. Vous pouvez également spécifier le paramètre **-i** sur la ligne de commande pour arrêter immédiatement un agent. Une fois l'agent arrêté, vous ne pouvez pas l'utiliser pour transférer des fichiers tant que vous ne l'avez pas redémarré.

### Référence associée

[Agent fteCreate\(créer un agent MFT \)](#)

[fteCreateCDAgent \(créer un agent de pont Connect:Direct \)](#)

[fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT \)](#)

[Agent fteModify\(exécutez un agent MFT en tant que service Windows \)](#)

### Information associée

[Le fichier MFT agent.properties](#)

## Linux AIX Démarrage d'un agent MFT au démarrage du système AIX and Linux

Un Managed File Transfer Agent peut être configuré pour démarrer au démarrage du système sur AIX and Linux. Lorsque vous vous déconnectez, votre agent continue de s'exécuter et peut recevoir des transferts de fichiers.

Une fois que vous avez créé et configuré un agent à l'aide de l'une des commandes Managed File Transfer suivantes: **fteCreateAgent**, **fteCreateCDAgent** ou **fteCreateBridgeAgent**, vous pouvez

le configurer pour qu'il démarre automatiquement lors d'un réamorçage sur les machines AIX and Linux à l'aide d'un fichier script qui exécute simplement la commande suivante:

```
su -l mqmft_user -c mq_install_root/bin/fteStartAgent agent_name
```

Où *mq\_install\_root* est le répertoire racine de l'installation Managed File Transfer requise, la valeur par défaut est: */opt/mqm* et *nom\_agent* est le nom du Managed File Transfer Agent à démarrer. L'utilisation de ce fichier script varie en fonction du système d'exploitation spécifique. Par exemple, des options supplémentaires sont disponibles sous Linux.

## Linux

### Linux

Pour les systèmes Linux , vous pouvez démarrer des applications de plusieurs manières au cours du processus d'amorçage du système. En général, tenez compte des étapes suivantes:

1. Créez un fichier appelé */etc/rc.mqmft* avec son contenu:

```
#!/bin/sh
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

Où *mqmft\_user* est l'ID utilisateur sous lequel le processus d'agent doit s'exécuter. Cet ID utilisateur doit être membre du groupe *mqm*.

2. Rendez le fichier exécutable, par exemple:

```
chmod 755 /etc/rc.mqmft
```

3. Ajoutez ensuite la ligne suivante à */etc/inittab*:

```
mqmft:5:boot:/etc/rc.mqmft
```

Vous pouvez également démarrer un agent lors de l'amorçage sur Linux en ajoutant les lignes de script au fichier */etc/rc.d/rc.local* ou sur Linux SuSe, en ajoutant les lignes de script au fichier */etc/init.d/boot.local* . Vous devez sélectionner la méthode qui convient le mieux à votre environnement. Voici quelques informations supplémentaires sur les autres méthodes de démarrage d'un agent lors du démarrage sur des distributions Linux spécifiques prises en charge:

### SLES 10 et 11

Pour les systèmes SUSE Linux Enterprise Server (SLES) 10 et 11, procédez comme suit:

1. En tant qu'ID utilisateur root du système, créez votre propre fichier */etc/init.d/rc.rclocal* .
2. Ajoutez les lignes suivantes au fichier *rc.rclocal* :

```
#!/bin/sh
### BEGIN INIT INFO
# Provides: rc.rclocal
# Required-Start: $network $syslog
# Required-Stop: $network $syslog
# Default-Stop: 0 1 2 6
# Description: MQMFT agent startup
### END INIT INFO
su -l mqmft_user"-c mq_install_root/bin/fteStartAgent agent_name"
```

3. Exécutez les commandes suivantes :

```
chmod 755 rc.rclocal
chkconfig --add rc.rclocal
```



## Démarrage des agents Managed File Transfer sous Linux avec systemd

Linux

Effectuez la procédure suivante :

1. Créez un fichier dans le dossier système `/etc/systemd/system/` et nommez-le, par exemple, `<agentname>.service`. Ajoutez le contenu suivant, où `<agentname>` correspond à `MFT_AGT_LNX_0`.

```
# vi /etc/systemd/system/MFT_AGT_LNX_0.service
[Unit]
Description=IBM MQ MFT MFT_AGT_LNX_0
[Service]
ExecStart=/opt/mqm/bin/fteStartAgent MFT_AGT_LNX_0
ExecStop=/opt/mqm/bin/fteStopAgent MFT_AGT_LNX_0
Type=forking
User=mqm
Group=mqm
KillMode=none
```

2. Pour activer le service, exécutez les commandes suivantes:

```
# systemctl enable MFT_AGT_LNX_0
# systemctl daemon-reload
```

3. Pour démarrer l'agent et vérifier son statut, exécutez les commandes suivantes:

```
# systemctl start MFT_AGT_LNX_0
# systemctl status MFT_AGT_LNX_0
```

### Tâches associées

«Arrêt d'un agent MFT», à la page 226

Vous pouvez arrêter un agent Managed File Transfer à partir de la ligne de commande. Lorsque vous arrêtez un agent, vous mettez l'agent au repos et vous l'autorisez à terminer son transfert de fichiers en cours avant de l'arrêter. Vous pouvez également spécifier le paramètre `-i` sur la ligne de commande pour arrêter immédiatement un agent. Une fois l'agent arrêté, vous ne pouvez pas l'utiliser pour transférer des fichiers tant que vous ne l'avez pas redémarré.

### Référence associée

[fteCreateAgent](#)

[fteCreateCDAgent](#): créez un agent de pont Connect:Direct

[fteCreateBridgeAgent](#) (créer et configurer un agent de pont de protocole MFT)

## z/OS Starting an MFT agent on z/OS

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

**Note:** Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

As a Java application, an agent is a z/OS UNIX System Services application that you can run from JCL by using the BFGAGSTP member, from a generated Managed File Transfer command PDSE library data set for an agent. For more information about how to create an MFT command PDSE library data set, and customize it for the required agent, see [Creating an MFT Agent or Logger command data set](#).

### Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

### Related reference

[“Stopping an MFT agent on z/OS” on page 227](#)

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

[The MFT agent.properties file](#)

## Liste des agents MFT

Vous pouvez répertorier les agents Managed File Transfer enregistrés auprès d'un gestionnaire de files d'attente particulier à l'aide de la ligne de commande ou de l' IBM MQ Explorer.

### Pourquoi et quand exécuter cette tâche

Pour répertorier les agents à l'aide de la ligne de commande, voir [CommandefteListAgents](#).

Pour répertorier les agents à l'aide de l' IBM MQ Explorer, dans la vue Navigator , cliquez sur **Agents** sous le nom du gestionnaire de files d'attente de coordination.

Si un agent n'est pas répertorié par la commande **fteListAgents** ou n'est pas affiché dans le IBM MQ Explorer, utilisez le diagramme de diagnostic de la rubrique suivante pour localiser et résoudre le problème: [Que faire si votre agent MFT n'est pas répertorié par la commande fteListAgents](#).

### Référence associée

[Agents fteList: répertorie les agents MFT pour un gestionnaire de files d'attente de coordination](#)

[Valeurs de statut de l'agent MFT](#)

[fteShowAgentDetails](#)

## Arrêt d'un agent MFT

Vous pouvez arrêter un agent Managed File Transfer à partir de la ligne de commande. Lorsque vous arrêtez un agent, vous mettez l'agent au repos et vous l'autorisez à terminer son transfert de fichiers en cours avant de l'arrêter. Vous pouvez également spécifier le paramètre **-i** sur la ligne de commande pour arrêter immédiatement un agent. Une fois l'agent arrêté, vous ne pouvez pas l'utiliser pour transférer des fichiers tant que vous ne l'avez pas redémarré.

### Avant de commencer

Si vous souhaitez vérifier les noms des agents associés à un gestionnaire de files d'attente, vous pouvez répertorier les agents à l'aide de la IBM MQ Explorer ou de la ligne de commande. Voir [Commande des agents fteList](#).

### Pourquoi et quand exécuter cette tâche

Pour arrêter un agent à partir de la ligne de commande, voir [fteStopAgent](#).

Si un agent est arrêté de manière contrôlée à l'aide de **fteStopAgent**, il n'accepte aucune nouvelle demande de transfert géré et attend que les transferts en cours soient terminés avant de s'arrêter. Depuis la IBM MQ 9.3.0, pour indiquer que l'agent est toujours dans un état transitoire et qu'il ne s'est

donc pas encore arrêté et qu'il ne peut pas encore être redémarré, l'agent passe à l'état STOPPING jusqu'à ce que les transferts en cours soient terminés. Ce statut apparaît dans la sortie des commandes **fteListAgents** et **fteShowAgentDetails**, dans [Requêtes MFT REST API](#) et dans la vue **Agents** du plug-in MFT de IBM MQ Explorer.

**Windows** Si vous avez configuré votre agent pour qu'il s'exécute en tant que service Windows, l'exécution de la commande **fteStopAgent** arrête également le service Windows. Vous pouvez également arrêter l'agent en arrêtant le service à l'aide de l'outil Windows Services. Pour plus d'informations, voir [«Démarrage d'un agent MFT en tant que service Windows»](#), à la page 221.

### Tâches associées

[«Démarrage d'un agent MFT»](#), à la page 220

Avant de pouvoir utiliser un agent Managed File Transfer pour un transfert de fichier, vous devez d'abord démarrer l'agent.

### Référence associée

[Valeurs de statut de l'agent MFT](#)

[fteStopAgent](#)

[«Stopping an MFT agent on z/OS»](#), à la page 227

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

## **z/OS** Stopping an MFT agent on z/OS

If you are running a Managed File Transfer Agent on z/OS as a started task from JCL, the agent accepts the z/OS operator commands **MODIFY** and **STOP**, in addition to the **fteStopAgent** command.

A started task is used because it runs under a specific user ID and is not affected by users logging off.

**Note:** Started tasks are typically run under an administrative user that might not have log-on privileges and so it is not possible to log on to the z/OS system as the user that the agent is running under. The **fteStartAgent**, **fteStopAgent**, **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, cannot be issued for that agent.

You can use the agent property **adminGroup** with Managed File Transfer agents on z/OS. You can define a security manager group, for example MFTADMIN and then add the started task userid and administrator TSO ids to this group. Edit the agent properties file and set the **adminGroup** property to be the name of this security manager group.

```
adminGroup=MFTADMIN
```

Members of this group can then issue the **fteStartAgent**, **fteStopAgent**, and **fteSetAgentTraceLevel** commands, and the **fteShowAgentDetails** command with the **-d** parameter specified, for the agent that is running as a started task.

For more information, see the **adminGroup** property in [The MFT agent.properties file](#).

## Controlled agent shutdown by using the z/OS MODIFY command (F)

The **MODIFY** command allows you to stop an agent in a controlled way as an alternative to the **fteStopAgent** command. The agent completes any transfers currently in progress but the agent does not start any new transfers.

For example:

```
F job_name,APPL=STOP
```

where *job\_name* is the job that the agent process is running under.

## Immediate agent shutdown by using the z/OS STOP command (P)

The **STOP** command is equivalent to an immediate stop by using the **fteStopAgent** command with the **-i** parameter. The agent is stopped immediately even if the agent is currently transferring a file.

For example:

```
P job_name
```

where *job\_name* is the job that the agent process is running under.

### Related concepts

[Enabling MFT agents to connect to remote z/OS queue managers](#)

### Related reference

[“Starting an MFT agent on z/OS” on page 225](#)

On z/OS, in addition to running the **fteStartAgent** command from a z/OS UNIX System Services session, you can start an agent as a started task from JCL without the need for an interactive session.

The MFT agent [.properties](#) file

## Démarrage d'un nouveau transfert de fichiers

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Pourquoi et quand exécuter cette tâche

Pour démarrer un nouveau transfert de fichier à partir de la ligne de commande, voir [fteCreateTransfer](#) command.

Pour démarrer un nouveau transfert de fichier à l'aide de l'assistant **Créer un transfert de fichier géré** dans IBM MQ Explorer, procédez comme suit:

### Procédure

1. Dans la vue Navigator , cliquez sur **Transfert de fichier géré. Managed File Transfer Central** s'affiche dans la vue Contenu.
2. Tous vos gestionnaires de files d'attente de coordination sont affichés dans la vue Navigator . Développez le nom du gestionnaire de file d'attente de coordination sur lequel l'agent que vous souhaitez utiliser pour le transfert est enregistré. Si vous êtes actuellement connecté à un gestionnaire de files d'attente de coordination autre que celui que vous souhaitez utiliser pour le transfert, cliquez avec le bouton droit de la souris sur le nom du gestionnaire de files d'attente de coordination dans la vue Navigator , puis cliquez sur **Déconnexion**. Cliquez ensuite avec le bouton droit de la souris sur le nom du gestionnaire de files d'attente de coordination à utiliser, puis cliquez sur **Connexion**.
3. Démarrez l'assistant **Créer un nouveau transfert de fichier géré** à l'aide de l'une des méthodes suivantes:
  - a) Cliquez avec le bouton droit de la souris sur le nom de l'un des noeuds suivants dans la vue Navigator : le gestionnaire de files d'attente de coordination approprié, **Modèles de transfert**, **Journal de transfert** ou **Transferts en attente**. Cliquez ensuite sur **Nouveau transfert** pour démarrer l'assistant.
  - b) Cliquez sur **Fichier > Nouveau > Autres > Assistants Managed File Transfer > Assistant Nouveau transfert**
4. Suivez les instructions des panneaux de l'assistant. Une aide contextuelle est également fournie pour chaque panneau. Pour accéder à l'aide contextuelle sur Windows, appuyez sur F1. Sous Linux, appuyez sur Ctrl+F1 ou sur Shift+F1.

### Concepts associés

[«Utilisation de fichiers de définition de transfert», à la page 229](#)

Vous pouvez spécifier un fichier de définition de transfert qui peut être utilisé pour créer un transfert de fichier. Le fichier de définition de transfert est un fichier XML qui définit tout ou partie des informations requises pour créer le transfert.

### Tâches associées

«Création d'un transfert de fichiers planifié», à la page 231

Vous pouvez planifier un nouveau transfert de fichier à partir du IBM MQ Explorer ou de la ligne de commande. Le transfert planifié peut contenir un seul fichier ou plusieurs fichiers dans un groupe. Vous pouvez effectuer un transfert de fichier planifié une seule fois ou répéter le transfert plusieurs fois.

«Déclenchement d'un transfert de fichiers», à la page 233

Vous pouvez définir certaines conditions de déclenchement sur un transfert de fichier qui doivent être vraies pour que ce transfert puisse avoir lieu. Si les conditions de déclenchement ne sont pas remplies, le transfert de fichier n'a pas lieu et un message de journal est éventuellement envoyé pour enregistrer le fait que le transfert n'a pas eu lieu. La demande de transfert de fichier est ensuite supprimée. Par exemple, vous pouvez configurer un transfert de fichier qui n'a lieu que si un fichier nommé sur le système où se trouve l'agent source est supérieur à une taille spécifiée ou s'il existe un fichier nommé particulier sur le système où se trouve l'agent source. Vous pouvez configurer un transfert de fichier déclenché à partir de IBM MQ Explorer ou de la ligne de commande.

«Définition d'un délai d'attente pour la récupération des transferts bloqués», à la page 343

Vous pouvez définir un délai de récupération de transfert pour les transferts de fichiers bloqués qui s'applique à tous les transferts d'un agent source. Vous pouvez également définir un délai de récupération de transfert pour un transfert individuel. Si vous définissez une durée spécifique, en secondes, pendant laquelle un agent source continue de tenter de récupérer un transfert de fichier bloqué et que le transfert échoue lorsque l'agent atteint le délai d'attente, le transfert échoue.

### Référence associée

**fteCreateTransfer**: démarrer un nouveau transfert de fichier

[Format de message de demande de transfert de fichier](#)

[Instructions pour le transfert de fichiers](#)

## Utilisation de fichiers de définition de transfert

Vous pouvez spécifier un fichier de définition de transfert qui peut être utilisé pour créer un transfert de fichier. Le fichier de définition de transfert est un fichier XML qui définit tout ou partie des informations requises pour créer le transfert.

Les fichiers de définition de transfert sont utiles lorsque vous souhaitez spécifier plusieurs fichiers source et plusieurs fichiers de destination dans une seule opération de transfert. Vous pouvez utiliser un fichier de définition de transfert pour soumettre un transfert de fichier complexe. Vous pouvez réutiliser et partager le fichier de définition de transfert.

Vous pouvez utiliser deux formats pour un fichier de définition de transfert et, bien que ces formats varient légèrement, ils sont tous deux conformes au schéma `FileTransfer.xsd`. Ce schéma se trouve dans le répertoire `samples\schema` de l'installation Managed File Transfer.

Les deux formats suivants de fichiers de définition de transfert sont pris en charge:

- Définition des fichiers source et cible pour un transfert. Cette définition utilise un élément **transferSpecifications** comme racine.
- Définition de l'ensemble du transfert, y compris les fichiers source et de destination et les agents source et de destination. Cette définition utilise un élément **request** comme racine.
  - Les fichiers de ce format peuvent être générés à partir de la commande **fteCreateTransfer** à l'aide du paramètre **-gt**.

L'exemple suivant illustre un format de fichier de définition de transfert qui spécifie uniquement les fichiers source et cible d'un transfert:

```
<?xml version="1.0" encoding="UTF-8"?>
<transferSpecifications xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
```

```

<item checksumMethod="MD5" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="directory" exist="overwrite">
    <file>c:\targetfiles</file>
  </destination>
</item>
</transferSpecifications>

```

Pour soumettre ce format de fichier de définition de transfert, vous devez spécifier les agents source et cible sur la ligne de commande:

```

fteCreateTransfer -sa AGENT1 -sm agent1qm -da AGENT2 -dm agent2qm -td
c:\definitions\example1.xml

```

L'exemple suivant illustre un format de fichier de définition de transfert qui spécifie toutes les informations requises pour un transfert:

```

<?xml version="1.0" encoding="UTF-8"?>
<request version="3.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>example.com.</hostName>
      <userID>fteuser</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="agent1qm"/>
    <destinationAgent agent="AGENT2" QMgr="agent2qm"/>
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:\sourcefiles\*.jpg</file>
        </source>
        <destination type="directory" exist="error">
          <file>/targetfiles/images</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

Vous pouvez générer un fichier avec ce format à l'aide du paramètre **-gt** de la commande **fteCreateTransfer**. Lorsque vous soumettez un fichier de définition de transfert avec ce format, vous n'avez pas besoin de spécifier autre chose sur la ligne de commande:

```

fteCreateTransfer -td c:\definitions\example2.xml

```

Vous pouvez remplacer les informations d'agent source et de destination sur la ligne de commande en transmettant les paramètres normaux en plus du fichier de définition de transfert. Exemple :

```

fteCreateTransfer -da AGENT9 -dm agent9qm -td c:\definitions\example2.xml

```

Cet exemple utilise les options de ligne de commande pour remplacer l'agent de destination défini dans le fichier de définition de transfert par **AGENT9** et le gestionnaire de files d'attente de destination défini dans le fichier de définition de transfert sous la forme **agent9qm**.

Les deux formats décrits peuvent contenir un ou plusieurs éléments < item>. Pour plus d'informations sur l'élément < item>, voir [Format de message de demande de transfert de fichier](#). Chacun de ces éléments de transfert définit une paire de fichiers source et de destination avec des attributs supplémentaires pour contrôler le comportement du transfert. Par exemple, vous pouvez spécifier le comportement suivant:

- Indique si le transfert utilise un total de contrôle
- Indique si le transfert est de type texte ou binaire
- Indique si le fichier source doit être supprimé une fois le transfert terminé
- Indique s'il faut remplacer le fichier de destination s'il existe

L'avantage de l'utilisation des fichiers de définition de transfert est que vous pouvez spécifier des options supplémentaires qui ne sont pas disponibles à partir de la ligne de commande. Par exemple, lorsque vous effectuez des transferts de message à fichier, vous pouvez spécifier l'attribut `groupId` à l'aide d'un fichier de définition de transfert. Cet attribut indique l'ID groupe IBM MQ des messages lus à partir de la file d'attente. Un autre avantage des fichiers de définition de transfert est que vous pouvez spécifier des options différentes pour chaque paire de fichiers. Par exemple, vous pouvez indiquer si un total de contrôle est utilisé ou si le fichier est transféré en mode texte ou binaire, fichier par fichier. Si vous utilisez la ligne de commande, les mêmes options s'appliquent à chaque fichier d'un transfert.

Exemple :

```
<item checksumMethod="none" mode="binary">
  <source disposition="leave">
    <file>c:\sourcefiles\source1.doc</file>
  </source>
  <destination type="file" exist="error">
    <file>c:\destinationfiles\destination1.doc</file>
  </destination>
</item>

<item checksumMethod="MD5" mode="text">
  <source disposition="delete">
    <file>c:\sourcefiles\source2.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file encoding="UTF8" EOL="CRLF">c:\destinationfiles\destination2.txt</file>
  </destination>
</item>

<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>c:\originfiles\source3.txt</file>
  </source>
  <destination type="file" exist="overwrite">
    <file>c:\targetfiles\destination3.txt</file>
  </destination>
</item>
```

**z/OS** Vous pouvez utiliser des éléments pour transférer un fichier d'un système réparti vers un système z/OS :

```
z/OS
<item checksumMethod="none" mode="text">
  <source recursive="false" disposition="leave">
    <file>textTransferTest.txt</file>
  </source>
  <destination type="dataset" exist="overwrite">
    <file encoding="IBM-1047">//TEXT.TRANS.TEST</file>
  </destination>
</item>
```

**z/OS** Cet exemple transfère le fichier `textTransferTest.txt` de l'agent source vers l'ensemble de données `//TEXT.TRANS.TEST` sur l'agent cible en mode texte. Ce transfert convertit les données source du codage par défaut de l'agent source (aucun attribut de codage source n'est spécifié) en page de codes: IBM-1047.

## Création d'un transfert de fichiers planifié

Vous pouvez planifier un nouveau transfert de fichier à partir du IBM MQ Explorer ou de la ligne de commande. Le transfert planifié peut contenir un seul fichier ou plusieurs fichiers dans un groupe. Vous pouvez effectuer un transfert de fichier planifié une seule fois ou répéter le transfert plusieurs fois.

### Pourquoi et quand exécuter cette tâche

Vous pouvez configurer une planification de transfert de fichiers pour qu'elle se produise une fois ou aux intervalles suivants:

- Toutes les minutes
- Horaire
- Quotidien
- Hebdomadaire
- Mensuel
- Annuel

Vous pouvez ensuite spécifier les occurrences à arrêter aux points suivants:

- A une heure et une date définies
- Après un nombre défini d'occurrences

Vous pouvez également indiquer que les occurrences se poursuivent indéfiniment.

Si un transfert planifié s'exécute à la même heure chaque jour, utilisez l'attribut **adjustScheduleTimeForDaylightSaving** dans le fichier de propriétés de l'agent pour ajuster l'heure à laquelle la planification a lieu lorsque les horloges changent. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).

Pour créer un transfert de fichier planifié à l'aide de la ligne de commande, utilisez les paramètres de planification (**-tb, -ss, -oi, -of, -ocet -es**) pour la [commande de transfertfiteCreate](#).

Pour créer un transfert de fichier planifié à l'aide de l'assistant **Créer un transfert de fichier géré** dans IBM MQ Explorer, procédez comme suit:

## Procédure

1. Dans la vue Navigator, cliquez sur **Transfert de fichier géré. Managed File Transfer Central** s'affiche dans la vue Contenu.
2. Tous vos gestionnaires de files d'attente de coordination sont affichés dans la vue Navigator. Développez le nom du gestionnaire de file d'attente de coordination sur lequel l'agent que vous souhaitez utiliser pour le transfert est enregistré. Si vous êtes actuellement connecté à un gestionnaire de files d'attente de coordination autre que celui que vous souhaitez utiliser pour le transfert, cliquez avec le bouton droit de la souris sur le nom du gestionnaire de files d'attente de coordination dans la vue Navigator, puis cliquez sur **Déconnexion**. Cliquez ensuite avec le bouton droit de la souris sur le nom du gestionnaire de files d'attente de coordination à utiliser, puis cliquez sur **Connexion**.
3. Démarrez l'assistant **Créer un nouveau transfert de fichier géré** à l'aide de l'une des méthodes suivantes:
  - a) Cliquez avec le bouton droit de la souris sur le nom de l'un des noeuds suivants dans la vue Navigator : le gestionnaire de files d'attente de coordination approprié, **Modèles de transfert, Journal de transfert** ou **Transferts en attente**. Cliquez ensuite sur **Nouveau transfert** pour démarrer l'assistant.
  - b) Cliquez sur **Fichier > Nouveau > Autres > Assistants Managed File Transfer > Assistant Nouveau transfert**
4. Suivez les instructions des panneaux de l'assistant. Veillez à cocher la case **Activer le transfert de planification** et à entrer les détails de votre planification dans l'onglet **Planification**. Les transferts de fichiers planifiés démarrent dans la minute suivant l'heure de début de la planification, s'il n'y a aucun problème susceptible d'affecter le transfert. Par exemple, il peut y avoir des problèmes avec votre réseau ou votre agent qui empêchent le démarrage du transfert planifié. Une aide contextuelle est fournie pour chaque panneau. Pour accéder à l'aide contextuelle sur Windows, appuyez sur F1. Sous Linux, appuyez sur Ctrl+F1 ou sur Shift+F1.

## Résultats

Pour plus d'informations sur les messages impliqués dans les transferts de fichiers planifiés, voir [Formats de message de journal de transfert de fichiers planifiés](#).



## Utilisation des transferts de fichiers en attente

Vous pouvez afficher les transferts de fichiers planifiés en attente à partir du IBM MQ Explorer. La fenêtre **Transferts en attente** affiche tous les transferts en attente enregistrés auprès du gestionnaire de file d'attente de coordination auquel vous êtes actuellement connecté.


### Pourquoi et quand exécuter cette tâche

Pour afficher le statut d'un transfert de fichiers planifié qui n'a pas encore démarré, procédez comme suit:

### Procédure

1. Développez **Managed File Transfer** dans la vue Navigator . **Managed File Transfer Central** s'affiche dans la vue Contenu.
2. Tous vos gestionnaires de files d'attente de coordination sont affichés dans la vue Navigator . Développez le nom du gestionnaire de file d'attente de coordination que vous avez utilisé pour le transfert planifié. Si vous souhaitez modifier le gestionnaire de file d'attente de coordination auquel vous êtes connecté, cliquez avec le bouton droit de la souris sur le nom du gestionnaire de file d'attente de coordination à utiliser dans la vue Navigator , puis cliquez sur **Connexion**.
3. Cliquez sur **Transferts en attente**. La fenêtre **Transferts en attente** s'affiche dans la vue Contenu.
4. La fenêtre **Transferts en attente** affiche les détails suivants sur vos transferts de fichiers planifiés:
  - a) **Nom** Numéro du transfert de fichiers planifié. Ce numéro est automatiquement attribué.
  - b) **Source** Nom de l'agent source.
  - c) **Fichier source** Nom du fichier à transférer sur son système hôte.
  - d) **Destination** Nom de l'agent de destination.
  - e) **Fichier de destination** Nom du fichier après son transfert vers le système de destination.
  - f) **Début planifié (fuseau horaire sélectionné)** Date et heure auxquelles le transfert de fichier est planifié pour démarrer dans le fuseau horaire sélectionné de l'administrateur. Pour modifier le fuseau horaire affiché, cliquez sur **Fenêtre > Préférences > IBM MQ Explorer > Managed File Transfer** et sélectionnez un autre fuseau horaire dans la liste **Fuseau horaire:** . Cliquez sur **OK**.
  - g) **Répéter toutes les** Si vous avez choisi de répéter le transfert planifié, l'intervalle spécifié que vous souhaitez répéter le transfert, exprimé sous la forme d'un nombre.
  - h) **Type de répétition** Si vous avez choisi de répéter le transfert planifié, le type d'intervalle de répétition que vous avez spécifié pour le transfert de fichiers. Le type peut être l'une des valeurs suivantes: minutes, heures, jours, semaines, mois ou années.
  - i) **Répéter jusqu'à** Si vous avez choisi de répéter le transfert planifié, les détails du moment où vous souhaitez que le transfert de fichiers répétitif s'arrête. Par exemple, une date et une heure spécifiées, ou après un nombre spécifié d'occurrences.

### Résultats

Pour actualiser les éléments affichés dans la fenêtre **Transferts en attente** , cliquez sur le bouton Actualiser  dans la barre d'outils de la vue Contenu.

Pour annuler un transfert de fichiers en attente, cliquez avec le bouton droit de la souris sur le transfert concerné et cliquez sur **Annuler**. L'annulation d'un transfert supprime complètement la demande de transfert de fichier.

## Déclenchement d'un transfert de fichiers

Vous pouvez définir certaines conditions de déclenchement sur un transfert de fichier qui doivent être vraies pour que ce transfert puisse avoir lieu. Si les conditions de déclenchement ne sont pas remplies, le transfert de fichier n'a pas lieu et un message de journal est éventuellement envoyé pour enregistrer le fait que le transfert n'a pas eu lieu. La demande de transfert de fichier est ensuite supprimée. Par

exemple, vous pouvez configurer un transfert de fichier qui n'a lieu que si un fichier nommé sur le système où se trouve l'agent source est supérieur à une taille spécifiée ou s'il existe un fichier nommé particulier sur le système où se trouve l'agent source. Vous pouvez configurer un transfert de fichier déclenché à partir de IBM MQ Explorer ou de la ligne de commande.

## Pourquoi et quand exécuter cette tâche

Vous pouvez surveiller une ressource en permanence pour qu'une condition de déclenchement soit satisfaite. Pour plus d'informations sur la surveillance des ressources, voir: [«Surveillance des ressources MFT»](#), à la page 239.

Vous pouvez définir trois conditions de déclenchement différentes. Les conditions sont les suivantes :

- Si un fichier particulier existe sur le même système que l'agent source
- Si un fichier particulier n'existe pas sur le même système que l'agent source
- Si un fichier particulier dépasse une certaine taille sur le système où se trouve l'agent source (la taille peut être exprimée en octets, ko, Mo ou Go). Ces unités de mesure utilisent la convention 2<sup>10</sup>, par exemple 1 ko est égal à 1024 octets et 1 Mo est égal à 1024 ko.

Les types de déclenchement de la liste précédente peuvent être combinés de deux manières:

- Pour une condition unique, vous pouvez spécifier plusieurs fichiers sur le système où se trouve l'agent source. Cela déclenche le transfert si l'un des fichiers spécifiés remplit la condition (opérateur booléen OR).
- Vous pouvez spécifier plusieurs conditions. Cela déclenche le transfert uniquement si toutes les conditions sont remplies (opérateur booléen AND).

Vous pouvez également combiner un transfert déclenché avec un transfert planifié. Pour plus d'informations, voir [Création d'un transfert de fichiers planifié](#). Dans ce cas, les conditions de déclenchement sont évaluées au moment où la planification doit démarrer, ou pour une planification répétitive chaque fois que la planification doit démarrer.

Les transferts déclenchés ne sont pas pris en charge sur les agents de pont de protocole.

Pour créer un transfert de fichiers déclenché à l'aide de la ligne de commande, utilisez le paramètre **-tr** dans la commande [fteCreateTransfer](#).

Pour créer un transfert de fichier planifié à l'aide de l'assistant **Créer un transfert de fichier géré** dans IBM MQ Explorer, procédez comme suit:

## Procédure

1. Dans la vue Navigator, cliquez sur **Transfert de fichier géré. Managed File Transfer Central** s'affiche dans la vue Contenu.
2. Tous vos gestionnaires de files d'attente de coordination sont affichés dans la vue Navigator. Développez le nom du gestionnaire de file d'attente de coordination que vous avez utilisé pour le transfert planifié. Si vous souhaitez modifier le gestionnaire de file d'attente de coordination auquel vous êtes connecté, cliquez avec le bouton droit de la souris sur le nom du gestionnaire de file d'attente de coordination à utiliser dans la vue Navigator, puis cliquez sur **Connexion**.
3. Démarrez l'assistant **Créer un nouveau transfert de fichier géré** à l'aide de l'une des méthodes suivantes:
  - a) Cliquez avec le bouton droit de la souris sur le nom de l'un des noeuds suivants dans la vue Navigator : le gestionnaire de files d'attente de coordination approprié, **Modèles de transfert**, **Journal de transfert** ou **Transferts en attente**. Cliquez ensuite sur **Nouveau transfert** pour ouvrir l'assistant.
  - b) Cliquez sur **Fichier > Nouveau > Autres > Assistants Managed File Transfer > Assistant Nouveau transfert**
4. Suivez les instructions des panneaux de l'assistant. Veillez à cocher la case **Activer le transfert déclenché** dans l'onglet **Déclencheurs** et à renseigner les zones de cet onglet pour configurer


le déclenchement. Une aide contextuelle est fournie pour chaque panneau. Pour accéder à l'aide contextuelle sur Windows, appuyez sur F1. Sous Linux, appuyez sur **Ctrl+F1** ou Shift+F1.

## Surveillance des transferts de fichiers en cours

Vous pouvez surveiller un transfert de fichiers en cours à l'aide de l'onglet **Managed File Transfer-Current Transfer Progress** dans IBM MQ Explorer. Ce transfert de fichier peut être démarré à partir de IBM MQ Explorer ou de la ligne de commande. L'onglet affiche également la progression des transferts planifiés au point de début des transferts planifiés.

### Pourquoi et quand exécuter cette tâche

Si vous souhaitez utiliser IBM MQ Explorer pour surveiller les transferts associés à un gestionnaire de files d'attente de coordination sur un système distant, suivez les instructions de la rubrique [«Configuration de IBM MQ Explorer pour la surveillance d'un gestionnaire de files d'attente de coordination distantes»](#), à la page 236 .

Les informations de transfert de fichiers précédentes ne sont pas conservées après l'arrêt et le redémarrage de IBM MQ Explorer. Au redémarrage, les informations sur les transferts passés sont effacées de l'onglet **Progression du transfert en cours** . Vous pouvez effacer les transferts terminés à l'aide de l'option **Supprimer les transferts terminés**  à tout moment lorsque IBM MQ Explorer est ouvert.


### Procédure


Une fois que vous avez démarré un nouveau transfert de fichiers à l'aide de IBM MQ Explorer ou de la ligne de commande, vous pouvez surveiller la progression de votre transfert dans l'onglet **Progression du transfert en cours** . Les informations suivantes s'affichent pour chaque transfert en cours:

- a) **Source**. Nom de l'agent utilisé pour transférer le fichier à partir du système source.
- b) **Destination**. Nom de l'agent utilisé pour recevoir le fichier sur le système cible.
- c) **Fichier en cours**. Nom du fichier en cours de transfert. La partie du fichier individuel qui a déjà été transférée est affichée en B, KiB, MiB, GiB ou TiB avec la taille totale du fichier entre parenthèses. L'unité de mesure affichée dépend de la taille du fichier.  
B est le nombre d'octets par seconde. KiB/s est le nombre de kibioctets par seconde, où 1 kibioctet est égal à 1024 octets. MiB/s correspond à des mébioctets par seconde, où 1 mébioctet est égal à 1 048 576 octets. GiB/s correspond à des gibioctets par seconde où 1 gibioctet est égal à 1 073 741 824 octets. TiB/s correspond à des tébioctets par seconde où 1 tébioctet est égal à 1 099 511 627 776 octets.
- d) **Numéro de fichier**. Si vous transférez plusieurs fichiers, ce nombre représente la distance par rapport au groupe total de fichiers transférés.
- e) **Progression**. La barre de progression indique le pourcentage d'achèvement du transfert de fichiers en cours.
- f) **Taux**. Débit de transfert du fichier dans KiB/s (kibioctets par seconde, où 1 kibioctet est égal à 1024 octets).
- g) **Démarré (fuseau horaire sélectionné)**. Heure à laquelle le transfert de fichier a commencé, présentée dans le fuseau horaire sélectionné de l'administrateur. Pour modifier le fuseau horaire affiché, cliquez sur **Fenêtre > Préférences > IBM MQ Explorer > Managed File Transfer** et sélectionnez un autre fuseau horaire dans la liste **Fuseau horaire** : . Cliquez sur **OK**.

Si le transfert passe à l'état de récupération lors du transfert du fichier, l'heure de début est mise à jour pour refléter l'heure à laquelle le transfert de fichier a repris.

## Résultats

Cet onglet actualise régulièrement ses informations automatiquement, mais pour forcer une vue actualisée de ce qui est affiché dans l'onglet **Progression du transfert en cours**, cliquez sur **Actualiser**  dans la barre d'outils de la vue de contenu.

Pour supprimer des transferts de fichiers à partir de l'onglet **Progression du transfert en cours**, cliquez sur **Supprimer les transferts terminés**  dans la barre d'outils de la vue Contenu. Cliquez sur ce bouton pour supprimer les détails du transfert de fichiers de l'onglet uniquement ; il n'arrête pas ou n'annule pas un transfert en cours ou planifié.

Si vous souhaitez revenir à l'onglet **Progression du transfert en cours** après sa fermeture, vous pouvez l'afficher en cliquant sur **Fenêtre > Afficher la vue > Autre > Autre > Transfert de fichiers géré-Progression du transfert en cours**. Cliquez sur **OK**.

## Que faire ensuite

De plus, il est possible de développer des applications pour la surveillance du transfert de fichiers personnalisé. Pour ce faire, vous pouvez créer un abonnement à la rubrique d'administration Managed File Transfer appropriée (à l'aide d'un programme ou d'un administrateur), et l'application de surveillance peut alors recevoir des publications d'activité de transfert de fichier Managed File Transfer sur la rubrique. Pour plus d'informations sur le format de message de publication et de rubrique d'abonnement, voir [Exemples de message de progression de transfert de fichier](#).

### Tâches associées

«[Configuration de IBM MQ Explorer pour la surveillance d'un gestionnaire de files d'attente de coordination distantes](#)», à la page 236

Utilisez IBM MQ Explorer pour surveiller les transferts de fichiers associés à un gestionnaire de files d'attente de coordination s'exécutant sur un système distant. Vous avez besoin d'un système capable d'exécuter le IBM MQ Explorer. Le composant IBM MQ Explorer doit être installé pour pouvoir se connecter au gestionnaire de files d'attente de coordination à distance.

«[Affichage du statut des transferts de fichiers dans le journal de transfert](#)», à la page 237

Vous pouvez afficher les détails des transferts de fichiers à l'aide du **Journal de transfert** dans IBM MQ Explorer. Il peut s'agir de transferts démarrés à partir de la ligne de commande ou du IBM MQ Explorer. Vous pouvez également personnaliser les éléments affichés dans le **Journal de transfert**.

## Configuration de IBM MQ Explorer pour la surveillance d'un gestionnaire de files d'attente de coordination distantes

Utilisez IBM MQ Explorer pour surveiller les transferts de fichiers associés à un gestionnaire de files d'attente de coordination s'exécutant sur un système distant. Vous avez besoin d'un système capable d'exécuter le IBM MQ Explorer. Le composant IBM MQ Explorer doit être installé pour pouvoir se connecter au gestionnaire de files d'attente de coordination à distance.

## Pourquoi et quand exécuter cette tâche

Hypothèses: droits permettant de se connecter au gestionnaire de files d'attente de coordination distantes en configurant le gestionnaire de files d'attente de manière à autoriser les connexions distantes.

Pour plus d'informations sur la configuration, voir [Connexion à un gestionnaire de files d'attente en mode client avec authentification de canal](#) et [Gestion des autorités pour les ressources spécifiques à MFT](#).

Pour surveiller les gestionnaires de files d'attente et les transferts de fichiers entre les agents sur un système qui n'exécute pas Windows ou Linux, configurez IBM MQ Explorer pour qu'il se connecte au système distant en procédant comme suit:

## Procédure

1. Démarrez le IBM MQ Explorer local.

2. Lorsque IBM MQ Explorer est chargé, cliquez avec le bouton droit de la souris sur le dossier **Managed File Transfer** et sélectionnez **Nouvelle configuration**.
3. Exécutez l'assistant en sélectionnant le gestionnaire de files d'attente de coordination et de commandes, puis définissez un nom pour la configuration.
4. Cliquez sur **Terminer** pour terminer la définition.
5. Une fois la définition terminée, cliquez avec le bouton droit de la souris sur la définition et sélectionnez **Connecter**.

## Résultats

A présent, démarrez IBM MQ Explorer et utilisez-le pour surveiller l'activité de transfert pour le réseau Managed File Transfer associé au gestionnaire de files d'attente de coordination.

### Tâches associées

«Surveillance des transferts de fichiers en cours», à la page 235

Vous pouvez surveiller un transfert de fichiers en cours à l'aide de l'onglet **Managed File Transfer-Current Transfer Progress** dans IBM MQ Explorer. Ce transfert de fichier peut être démarré à partir de IBM MQ Explorer ou de la ligne de commande. L'onglet affiche également la progression des transferts planifiés au point de début des transferts planifiés.


«Affichage du statut des transferts de fichiers dans le journal de transfert», à la page 237

Vous pouvez afficher les détails des transferts de fichiers à l'aide du **Journal de transfert** dans IBM MQ Explorer. Il peut s'agir de transferts démarrés à partir de la ligne de commande ou du IBM MQ Explorer. Vous pouvez également personnaliser les éléments affichés dans le **Journal de transfert**.

## Affichage du statut des transferts de fichiers dans le journal de transfert

Vous pouvez afficher les détails des transferts de fichiers à l'aide du **Journal de transfert** dans IBM MQ Explorer. Il peut s'agir de transferts démarrés à partir de la ligne de commande ou du IBM MQ Explorer. Vous pouvez également personnaliser les éléments affichés dans le **Journal de transfert**.

### Procédure



1. Développez **Managed File Transfer** dans la vue Navigator , puis développez le nom du gestionnaire de files d'attente de coordination dont vous souhaitez afficher le journal de transfert.
2. Cliquez sur **Journal de transfert** dans la vue Navigator . Le **journal de transfert** s'affiche dans la vue Contenu.
3. La fenêtre **Journal de transfert** affiche les détails suivants sur vos transferts de fichiers:
  - a) **Source** Nom de l'agent sur le système où se trouve le fichier source.
  - b) **Destination** Nom de l'agent sur le système vers lequel vous souhaitez transférer le fichier.
  - c) **Etat d'achèvement** Statut du transfert de fichier. L'état peut être l'une des valeurs suivantes: "Démarré", "En cours", "Réussite", "Partiellement réussi", "Annulé" ou "Echec".
  - d) **Propriétaire** ID utilisateur sur l'hôte qui a soumis la demande de transfert.
  - e) **Démarré (fuseau horaire sélectionné)** Date et heure auxquelles la demande de transfert de fichier a été acceptée par l'agent Managed File Transfer , présentées dans le fuseau horaire sélectionné de l'administrateur. Pour modifier le fuseau horaire affiché, cliquez sur **Fenêtre > Préférences > IBM MQ Explorer > Managed File Transfer** et sélectionnez un autre fuseau horaire dans la liste **Fuseau horaire** : . Cliquez sur **OK**.
  - f) **Etat enregistré (fuseau horaire sélectionné)** (Cette colonne n'est pas affichée par défaut. Vous pouvez choisir d'afficher la colonne à l'aide de la fenêtre **Configurer les colonnes du journal de transfert**  . Heure et date d'enregistrement de l'état d'achèvement, dans le fuseau horaire sélectionné par l'administrateur.
  - g) **Nom du travail** Identificateur spécifié par l'utilisateur à l'aide du paramètre **-jn** de **fteCreateTransfer** ou dans un script Ant

- h) **ID transfert** Identificateur unique du transfert de fichier.
- i) **Connect: Direct** Les détails relatifs au **numéro de processus**, au **nom de processus**, au **noeud principal**, au **noeud secondaire**, au **type de source** et au **type de destination** sont répertoriés.

## Résultats

**Remarque :** Le format interne du journal de transfert a été modifié dans IBM MQ 8.0.0 Fix Pack 1 pour l'APAR IC99545. Par conséquent, si un IBM MQ Explorer est mis à niveau vers V8.0.0.1 ou une version ultérieure, puis restauré dans V8.0.0.0, aucun XML d'audit n'est affiché pour les transferts qui ont eu lieu alors que IBM MQ Explorer était à V8.0.0.1. Le panneau XML de la fenêtre **Propriétés** de ces transferts contient une zone de texte vide.

Pour afficher plus de détails sur un transfert terminé, développez le transfert qui vous intéresse en cliquant sur le signe plus (+). Vous pouvez ensuite voir tous les noms de fichier source et de destination inclus dans ce transfert. Toutefois, si le transfert est en cours et se compose de nombreux fichiers, vous pouvez afficher uniquement les fichiers qui ont déjà été transférés jusqu'à présent.

Pour actualiser les éléments affichés dans le **Journal de transfert**, cliquez sur le bouton **Actualiser**  dans la barre d'outils de la vue de contenu. Les informations de transfert de fichier dans le journal de transfert restent dans le journal après l'arrêt et le redémarrage de IBM MQ Explorer. Si vous souhaitez supprimer du journal tous les transferts de fichiers terminés, cliquez sur **Supprimer les transferts terminés**  dans la barre d'outils de la vue Contenu.

Pour supprimer un transfert de fichier individuel terminé du journal, cliquez avec le bouton droit de la souris sur le transfert et cliquez sur **Supprimer**. Si vous supprimez un transfert, il n'arrête pas ou n'annule pas un transfert qui est en cours ou qui a été planifié ; vous supprimez uniquement les données d'historique stockées.

Pour copier l'identificateur unique d'un transfert dans le presse-papiers, cliquez avec le bouton droit de la souris sur ce transfert et cliquez sur **Copier l'ID**.

Les métadonnées et le XML d'audit complet du transfert sont disponibles dans le menu contextuel, sous l'action **Propriétés**.

### Tâches associées

«Surveillance des transferts de fichiers en cours», à la page 235

Vous pouvez surveiller un transfert de fichiers en cours à l'aide de l'onglet **Managed File Transfer-Current Transfer Progress** dans IBM MQ Explorer. Ce transfert de fichier peut être démarré à partir de IBM MQ Explorer ou de la ligne de commande. L'onglet affiche également la progression des transferts planifiés au point de début des transferts planifiés.

«Configuration du journal de transfert», à la page 238

Vous pouvez configurer les informations qui sont affichées et la façon dont les informations sont affichées dans le **journal de transfert** dans le IBM MQ Explorer.

«Définition d'un délai d'attente pour la récupération des transferts bloqués», à la page 343

Vous pouvez définir un délai de récupération de transfert pour les transferts de fichiers bloqués qui s'applique à tous les transferts d'un agent source. Vous pouvez également définir un délai de récupération de transfert pour un transfert individuel. Si vous définissez une durée spécifique, en secondes, pendant laquelle un agent source continue de tenter de récupérer un transfert de fichier bloqué et que le transfert échoue lorsque l'agent atteint le délai d'attente, le transfert échoue.


## Configuration du journal de transfert

Vous pouvez configurer les informations qui sont affichées et la façon dont les informations sont affichées dans le **journal de transfert** dans le IBM MQ Explorer.


## Pourquoi et quand exécuter cette tâche

Pour réorganiser l'ordre des colonnes dans le **Journal de transfert**, cliquez sur le titre de la colonne à déplacer et faites glisser la colonne vers sa nouvelle position. Le nouvel ordre des colonnes est conservé uniquement jusqu'à ce que vous arrêtez et redémarriez IBM MQ Explorer.

Pour filtrer les entrées du **journal de transfert**, entrez une chaîne dans la zone **Filtrer les entrées de journal affichées**. Pour restaurer toutes les entrées du journal, supprimez la chaîne que vous avez entrée dans la zone. Vous pouvez utiliser n'importe quelle expression régulière Java valide dans cette zone. Pour plus d'informations, voir [Expressions régulières utilisées par MFT](#).

Pour personnaliser les colonnes affichées dans le journal de transfert, utilisez **Configurer les colonnes du journal de transfert** . Procédez comme suit pour démarrer et utiliser la fenêtre **Configurer les colonnes du journal de transfert**.

## Procédure

1. Vérifiez que le **journal de transfert** est ouvert dans la vue Contenu. Cliquez sur **Configurer les colonnes du journal de transfert**  dans la barre d'outils de la vue de contenu. La fenêtre **Configurer les colonnes du journal de transfert** s'ouvre.
2. Pour personnaliser votre vue du **journal de transfert**, cochez ou décochez des cases individuelles pour les colonnes que vous souhaitez afficher ou masquer. Vous pouvez cliquer sur **Sélectionner tout**, puis sur **OK** pour sélectionner toutes les cases à cocher ou sur **Désélectionner tout**, puis sur **OK** pour désélectionner toutes les cases.

## Tâches associées

[«Surveillance des transferts de fichiers en cours»](#), à la page 235

Vous pouvez surveiller un transfert de fichiers en cours à l'aide de l'onglet **Managed File Transfer-Current Transfer Progress** dans IBM MQ Explorer. Ce transfert de fichier peut être démarré à partir de IBM MQ Explorer ou de la ligne de commande. L'onglet affiche également la progression des transferts planifiés au point de début des transferts planifiés.

[«Affichage du statut des transferts de fichiers dans le journal de transfert»](#), à la page 237

Vous pouvez afficher les détails des transferts de fichiers à l'aide du **Journal de transfert** dans IBM MQ Explorer. Il peut s'agir de transferts démarrés à partir de la ligne de commande ou du IBM MQ Explorer. Vous pouvez également personnaliser les éléments affichés dans le **Journal de transfert**.

## Surveillance des ressources MFT

Vous pouvez surveiller les ressources Managed File Transfer ; par exemple, une file d'attente ou un répertoire. Lorsqu'une condition est satisfaite sur cette ressource, le moniteur de ressources démarre une tâche, telle qu'un transfert de fichier. Vous pouvez créer un moniteur de ressources à l'aide de la commande **fteCreateMonitor** ou de la vue **Moniteurs** dans le plug-in Managed File Transfer pour IBM MQ Explorer.

## Pourquoi et quand exécuter cette tâche

La surveillance des ressources Managed File Transfer utilise la terminologie suivante:

### Moniteur de ressources

Un moniteur de ressources est un processus qui interroge une ressource (telle qu'un répertoire ou une file d'attente) à un intervalle régulier prédéfini pour voir si le contenu de la ressource a été modifié. Si tel est le cas, le contenu est comparé à l'ensemble des conditions de ce moniteur. S'il existe une correspondance, la tâche de ce moniteur est démarrée.

### Ressource

La ressource système que le moniteur de ressources examine à chaque intervalle d'interrogation pour être comparée aux conditions de déclenchement. Les files d'attente, les répertoires ou les structures de répertoire imbriquées peuvent être la ressource contrôlée.

### **Condition et condition de déclenchement**

Une condition est une expression évaluée (généralement par rapport au contenu de la ressource surveillée). Si l'expression a pour résultat la valeur true, la condition contribue à la condition de déclenchement globale.

La condition de déclenchement est la condition globale, qui est satisfaite lorsque toutes les conditions sont remplies. Lorsque la condition de déclenchement est satisfaite, la tâche peut se poursuivre.

### **Tâche**

Une tâche est l'opération qui est démarrée lorsque la condition de déclenchement ou l'ensemble de conditions est satisfait. Les tâches prises en charge sont le transfert de fichiers et l'appel de commande.

### **Fichier de déclenchement**

Un fichier de déclenchement est un fichier placé dans un répertoire contrôlé pour indiquer qu'une tâche (généralement un transfert) peut commencer. Par exemple, il peut indiquer que tous les fichiers à traiter sont arrivés dans un emplacement connu et peuvent être transférés ou traités d'une autre manière. Le nom du fichier de déclenchement peut être utilisé pour spécifier les fichiers à transférer à l'aide de la substitution de variable. Pour plus d'informations, voir [«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables»](#), à la page 252.

Le fichier de déclenchement est également appelé fichier prêt ou fichier go. Toutefois, dans cette documentation, il est généralement appelé fichier de déclenchement.

La surveillance des ressources n'est pas prise en charge sur les agents de pont de protocole ou les agents de pont Connect:Direct .

### **Concepts associés**

[Conseils pour la configuration d'un moniteur de ressources MFT afin d'éviter la surcharge d'un agent](#)

### **Référence associée**

[fteCreateMonitor](#): créer un moniteur de ressources MFT

[fteListMoniteurs](#): répertorie les moniteurs de ressources MFT

[Moniteur fteDelete](#): suppression d'un moniteur de ressources MFT

[Formats de message de demande du moniteur MFT](#)

## **Concepts de surveillance des ressources MFT**

Présentation des concepts clés de la fonction de surveillance des ressources Managed File Transfer .

### **Des moniteurs de ressources**

Vous créez un moniteur de ressources à l'aide de la commande **fteCreateMonitor** , qui crée et démarre un nouveau moniteur de ressources à partir de la ligne de commande. Le moniteur de ressources est associé à un agent Managed File Transfer et n'est actif que lorsque cet agent est en cours d'exécution. Lorsque l'agent de surveillance s'arrête, il en est de même pour le moniteur de ressources. Si l'agent est déjà en cours d'exécution lors de la création du moniteur de ressources, ce dernier démarre immédiatement. L'agent de surveillance doit également être l'agent source de la tâche lancée par le moniteur de ressources.

Les noms de moniteur de ressources doivent être uniques au sein de leur agent. Le nom du moniteur de ressources doit comporter au moins un caractère et ne doit pas contenir d'astérisque (\*), de pourcentage (%) ou de point d'interrogation (?). La casse dans laquelle un nom de moniteur de ressources est fourni est ignorée et le nom de moniteur de ressources est converti en majuscules. Si vous tentez de créer un moniteur de ressources avec un nom déjà présent, la demande est ignorée et la tentative est consignée dans la rubrique du journal du moniteur de ressources.

**Remarque :** Vous ne pouvez pas créer de moniteur de ressources avec une définition de tâche contenant des transferts planifiés.

Avant IBM MQ 9.3.0, le seul moyen d'arrêter un moniteur de ressources consiste à arrêter l'agent qui exécute l'opération de moniteur. Pour redémarrer un moniteur de ressources, vous devez redémarrer l'agent. Depuis IBM MQ 9.3.0, vous pouvez démarrer et arrêter les moniteurs de ressources sans avoir



à arrêter ou à redémarrer un agent. Pour plus d'informations, voir [«Démarrage et arrêt des moniteurs de ressources»](#), à la page 243.

Il n'y a pas de restriction sur le nombre de moniteurs de ressources pouvant être créés sur un agent et tous exécutés avec la même priorité. Tenez compte des implications du chevauchement des ressources surveillées, des conditions de déclenchement conflictuelles et de la fréquence d'interrogation des ressources.

Le chevauchement des moniteurs de ressources peut entraîner:

- Conflit possible sur le / les emplacement (s) source (s).
- Demandes de transfert en double possibles pour les mêmes éléments source.
- Erreurs ou échecs inattendus pour les transferts dus à des conflits d'éléments source.

Si plusieurs moniteurs analysent le même emplacement et peuvent se déclencher sur les mêmes éléments, vous pouvez potentiellement vous retrouver avec le problème de deux moniteurs différents soumettant des demandes de transfert gérées pour le même élément.

Les moniteurs de ressources regardent le contenu des ressources après chaque période d'intervalle d'interrogation. Le contenu de la ressource est comparé aux conditions de déclenchement et si ces conditions sont satisfaites, la tâche associée au moniteur de ressources est appelée.

La tâche est démarrée de manière asynchrone. S'il existe une condition de correspondance et que la tâche est démarrée, le moniteur de ressources continue d'interroger les modifications apportées au contenu de la ressource. Par exemple, si une correspondance est trouvée parce qu'un fichier appelé `reports.go` est arrivé dans un répertoire contrôlé, la tâche est démarrée une seule fois. Lors de l'intervalle d'interrogation suivant, même si le fichier existe encore, la tâche n'est pas redémarrée. Toutefois, si le fichier est supprimé, puis placé à nouveau dans le répertoire, ou si le fichier est mis à jour (de sorte que l'attribut de date de dernière modification est modifié), la vérification de la condition de déclenchement suivante entraîne un nouvel appel de la tâche.

Avant IBM MQ 9.1.5, si un moniteur de ressources effectue une interrogation qui prend plus de temps que l'intervalle d'interrogation, cela signifie que l'interrogation suivante démarre dès que l'interrogation en cours se termine sans écart, ce qui peut avoir un impact sur la vitesse à laquelle les moniteurs de ressources soumettent le travail à un agent. Cela peut entraîner des problèmes de performances si les éléments trouvés lors de la première interrogation sont toujours présents lors de la seconde.

Le moniteur de ressources utilise le service `ScheduledExecutoret` lance l'interrogation suivante uniquement après la fin de l'interrogation précédente plus l'intervalle d'interrogation configuré. Cela signifie qu'il y aura toujours un écart entre les intervalles d'interrogation, plutôt que d'avoir une autre interrogation commençant immédiatement après l'interrogation précédente si le temps d'interrogation était supérieur à l'intervalle d'interrogation.

Si le transfert d'un fichier a échoué, vous pouvez effacer l'historique du moniteur de ressources, ce qui permet de soumettre une autre demande de transfert sans avoir à supprimer le fichier et à le replacer dans le répertoire, ou à mettre à jour le fichier pour modifier son attribut de date de dernière modification. L'effacement de l'historique est utile, par exemple, dans les cas où il est nécessaire que le fichier soit transféré, mais où la modification du fichier n'est pas possible. Pour plus d'informations, voir [«Effacement de l'historique du moniteur de ressources»](#), à la page 270.

## Ressources

Les moniteurs de ressources dans Managed File Transfer peuvent interroger le contenu des deux types de ressource suivants:

### Répertoires ou structures de répertoire imbriquées

Un scénario courant consiste à surveiller un répertoire pour détecter la présence d'un fichier de déclenchement. Une application externe peut traiter plusieurs fichiers et les placer dans un répertoire source connu. Lorsque l'application a terminé son traitement, elle indique que les fichiers sont prêts à être transférés, ou à être traités d'une autre manière, en plaçant un fichier de déclenchement dans un emplacement surveillé. Le fichier de déclenchement peut être détecté par un moniteur de ressources

Managed File Transfer et le transfert de ces fichiers du répertoire source vers un autre Managed File Transfer Agent est lancé.

Par défaut, le répertoire spécifié est surveillé. Pour examiner également les sous-répertoires, définissez le niveau de récursivité dans la commande **fteCreateTransfer** .

Voici deux exemples de surveillance d'un répertoire:

- Surveillez un fichier de déclenchement (par exemple, `trigger.file`), puis transférez un caractère générique (par exemple, `*.zip`).
- Surveillez `*.zip` , puis transférez `${FilePath}` (par exemple, le fichier qui a déclenché le transfert). Pour plus d'informations sur la substitution de variables, voir [«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables»](#), à la page 252.

**Remarque :** Ne créez pas de moniteur qui surveille `*.zip` , puis transfère `*.zip` . Le moniteur tente de démarrer un transfert de `*.zip` pour chaque fichier `.zip` de votre système. C'est-à-dire que le moniteur génère \* nombre de transferts pour `*.zip` .

Pour obtenir un exemple de création d'un moniteur de ressources afin de surveiller un répertoire, voir [«Surveillance d'un répertoire et utilisation de la substitution de variable»](#), à la page 249.

### Files d'attente IBM MQ

Un exemple de surveillance d'une file d'attente est qu'une application externe peut générer des messages et les placer dans une file d'attente connue avec le même ID de groupe. Lorsque l'application a terminé d'insérer des messages dans la file d'attente, elle indique que le groupe est terminé. Le groupe complet de messages peut être détecté par un moniteur de ressources Managed File Transfer et le transfert du groupe de messages de la file d'attente source vers un fichier est lancé. Pour un exemple de création d'un moniteur de ressources pour surveiller une file d'attente, voir [«Exemple: configuration d'une ressource MFT»](#), à la page 251.

**Remarque :** Vous ne pouvez spécifier qu'un seul moniteur par file d'attente. Si vous spécifiez plusieurs moniteurs pour interroger une file d'attente IBM MQ , un comportement imprévisible se produit.

Les fichiers de surveillance ne sont pas pris en charge.

### Conditions et conditions de déclenchement

La condition est remplie lorsque la ressource contient une valeur qui correspond à une autre chaîne ou à un autre modèle. Les conditions peuvent être l'une des suivantes:

- Correspondance sur le nom de fichier (modèle)
- Aucune correspondance sur le nom de fichier (modèle)
- Taille de fichier
- Correspond si la taille du fichier reste la même pour un certain nombre d'interrogations

La correspondance de nom de fichier peut être exprimée comme suit:

- Correspondance de chaîne exacte
- Correspondance avec des caractères génériques simples, comme décrit dans [Utilisation de caractères génériques avec MFT](#)
- Correspondance d'expression régulière

Les noms de fichier peuvent également être exclus de la mise en correspondance des noms de fichier à l'aide d'un caractère générique ou d'une expression régulière Java qui identifie les noms de fichier qui ne sont jamais mis en correspondance.

Lorsqu'un fichier correspondant est détecté, son horodatage de dernière modification est conservé. Si les interrogations suivantes détectent que le fichier a été modifié, la condition de déclenchement est remplie à nouveau et la tâche est démarrée. Si la condition est de détecter lorsqu'un fichier n'existe pas, si aucun fichier du répertoire contrôlé ne correspond au modèle de nom de fichier, la tâche est démarrée. Si un fichier est ensuite ajouté au répertoire qui correspond au modèle de nom de fichier, la tâche est démarrée uniquement si le fichier est ensuite supprimé.

## Tâches

Managed File Transfer prend en charge les deux types de tâche suivants que vous pouvez configurer pour qu'ils soient démarrés par les moniteurs de ressources:

### Tâches de transfert de fichiers

Les tâches de transfert de fichiers sont définies de la même manière que tout autre transfert de fichiers. Une méthode utile pour générer le code XML de tâche requis par un moniteur consiste à exécuter la commande `fteCreateTransfer` avec le paramètre `-gt`. Cette commande génère une définition de tâche en tant que document XML, y compris la spécification de transfert. Vous transmettez ensuite le nom du document XML de tâche comme valeur du paramètre `-mt` dans la commande `fteCreateMonitor`. Lorsque le **fteCreateMonitor** est exécuté, il lit le document XML de la tâche. Une fois le **fteCreateMonitor** exécuté, les modifications apportées au fichier XML de tâche ne sont pas utilisées par le moniteur.

Lorsque vous utilisez une tâche de transfert de fichiers, vous pouvez sélectionner le nombre de conditions de déclenchement qui sont regroupées par lots dans une tâche. Par défaut, une condition de déclenchement démarre une tâche. Vous pouvez exécuter la commande `fteCreateMonitor` avec l'option `-bs` pour sélectionner le nombre de conditions de déclenchement regroupées par lots dans une tâche.

### Tâches de commande

Les tâches de commande peuvent exécuter des scripts Ant, appeler des programmes exécutables ou exécuter des travaux JCL. Pour plus d'informations, voir [«Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts»](#), à la page 245.

## Fichiers de déclenchement

Vous pouvez utiliser le contenu d'un fichier de déclenchement dans un moniteur de ressources pour définir un ensemble de fichiers à transférer dans une demande de transfert unique. Chaque fois qu'un fichier de déclenchement correspondant est détecté, son contenu est analysé pour les chemins d'accès aux fichiers source et éventuellement pour les chemins d'accès aux fichiers de destination. Ces chemins de fichier sont ensuite utilisés pour définir des éléments de fichier dans le fichier XML de transfert de tâche que vous spécifiez, qui est soumis en tant que demande de transfert unique à l'agent. La définition du moniteur de ressources détermine si le contenu du déclencheur est activé.

Le format de chaque fichier de déclenchement est un chemin de fichier unique à transférer sur chaque ligne de texte. Le format par défaut de la ligne est soit un chemin de fichier source unique, soit un chemin de fichier source et de destination séparés par une virgule.

Pour plus d'informations et d'exemples, consultez [«Utilisation d'un fichier de déclenchement»](#), à la page 261.

## Démarrage et arrêt des moniteurs de ressources

Avant IBM MQ 9.3.0, le seul moyen d'arrêter un moniteur de ressources consiste à arrêter l'agent qui exécute l'opération de moniteur. Pour redémarrer un moniteur de ressources, vous devez redémarrer l'agent. Pour plus d'informations, reportez-vous aux sections [«Démarrage d'un agent MFT»](#), à la page 220 et [«Arrêt d'un agent MFT»](#), à la page 226.

Depuis la IBM MQ 9.3.0, vous pouvez démarrer et arrêter les moniteurs de ressources sans avoir à arrêter ou à redémarrer un agent à l'aide des commandes **fteStartMonitor** et **fteStopMonitor**. Ceci est utile, par exemple, dans les situations suivantes :

- Si un agent possède plusieurs moniteurs de ressources, et que seuls certains d'entre eux ont rencontré des erreurs, mais que les moniteurs de ressources restants fonctionnent toujours correctement, alors vous pouvez simplement redémarrer les moniteurs de ressources défectueux.
- Si vous souhaitez arrêter un moniteur de ressources pour effectuer des travaux de maintenance, ou si le moniteur de ressources n'est pas nécessaire pendant un certain temps et que vous ne voulez pas qu'il fonctionne inutilement, consommant ainsi de précieuses ressources système.

Pour plus d'informations, reportez-vous aux sections «Démarrage d'un moniteur de ressources MFT», à la page 266 et «Arrêt d'un moniteur de ressources MFT», à la page 267.

<i>Tableau 9. Comportement d'un moniteur de ressources en fonction de la commande exécutée</i>	
<b>Commande</b>	<b>Comportement du moniteur de ressources</b>
<b>fteStartMonitor</b>	Si l'agent est en cours d'exécution, le moniteur de ressources est démarré s'il est actuellement arrêté.
<b>fteStopMonitor</b>	Si l'agent est en cours d'exécution, le moniteur de ressources est arrêté s'il est actuellement démarré.
<b>fteStartAgent</b>	Le moniteur de ressources est démarré dans le cadre du démarrage de l'agent, quels que soient les appels antérieurs à <b>fteStopMonitor</b> .
<b>fteStopAgent</b>	Tous les moniteurs de ressources en cours d'exécution sont arrêtés.

## Sauvegarde et restauration des moniteurs de ressources

Vous pouvez sauvegarder les moniteurs de ressources que vous avez déjà définis afin de pouvoir les réutiliser ultérieurement. Il existe différentes options que vous pouvez utiliser comme suit:

- Utilisez la commande **fteCreateMonitor** avec le paramètre **-ox** pour exporter une configuration de moniteur de ressources dans un fichier XML et avec le paramètre **-ix** pour restaurer un moniteur de ressources en important la configuration de moniteur de ressources à partir d'un fichier XML.
- Utilisez la commande **fteListMonitors** avec **-ox** pour exporter la définition d'un moniteur de ressources unique dans un fichier XML.
- Utilisez la commande **fteListMonitors** avec **-od** pour exporter plusieurs définitions de moniteur de ressources dans un répertoire spécifié. Chaque définition de moniteur de ressources est sauvegardée dans un fichier XML distinct. Vous pouvez également utiliser l'option **-od** pour exporter une définition de moniteur de ressources unique vers un répertoire spécifié.

Pour plus d'informations, voir «Sauvegarde et restauration des moniteurs de ressources MFT», à la page 269.

## Consignation du moniteur de ressources

Managed File Transfer inclut la journalisation du moniteur de ressources. Pour plus d'informations, voir «Consignation des moniteurs de ressources MFT», à la page 264.

### Concepts associés

«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

### Tâches associées

«Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts», à la page 245

Les moniteurs de ressources ne sont pas limités à l'exécution de transferts de fichiers en tant que tâche associée. Vous pouvez également configurer le moniteur pour qu'il appelle d'autres commandes à partir de l'agent de surveillance, notamment des programmes exécutables, des scripts Ant ou des travaux JCL.

Pour appeler des commandes, éditez le fichier XML de définition de tâche de contrôle pour inclure un ou plusieurs éléments de commande avec les paramètres d'appel de commande correspondants, tels que les arguments et les propriétés.

«Exemple: configuration d'une ressource MFT», à la page 251

Vous pouvez spécifier une file d'attente IBM MQ comme ressource à surveiller par un moniteur de ressources à l'aide du paramètre **-mq** avec la commande **fteCreateMonitor**.

«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

### Référence associée

**fteCreateMonitor**: créer un moniteur de ressources MFT

**fteListMoniteurs**: répertorie les moniteurs de ressources MFT

**Moniteur fteDelete**: suppression d'un moniteur de ressources MFT

## Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts

Les moniteurs de ressources ne sont pas limités à l'exécution de transferts de fichiers en tant que tâche associée. Vous pouvez également configurer le moniteur pour qu'il appelle d'autres commandes à partir de l'agent de surveillance, notamment des programmes exécutables, des scripts Ant ou des travaux JCL. Pour appeler des commandes, éditez le fichier XML de définition de tâche de contrôle pour inclure un ou plusieurs éléments de commande avec les paramètres d'appel de commande correspondants, tels que les arguments et les propriétés.

### Pourquoi et quand exécuter cette tâche

Le chemin d'accès au programme exécutable, au script Ant ou au travail JCL que vous souhaitez que l'agent de surveillance appelle doit être inclus dans le fichier `commandPath` de l'agent de surveillance. Pour plus d'informations sur la propriété de chemin de commande, voir [commandPath MFT property](#).

Vous pouvez créer le document XML de définition de tâche de l'une des manières suivantes:

- Créez le document XML de définition de tâche manuellement en fonction du schéma `FileTransfer.xsd`.
- Utilisez un document XML généré comme base pour votre définition de tâche.

Que vous souhaitiez une tâche de transfert ou une tâche de commande, la définition de tâche doit commencer par un élément racine `<request>`. L'élément enfant de `<request>` doit être `<managedTransfer>` ou `<managedCall>`. Vous choisissez généralement `<managedCall>` lorsqu'il n'y a qu'une seule commande ou un seul script à exécuter, et `<managedTransfer>` si vous souhaitez que la tâche inclue un transfert de fichier et éventuellement jusqu'à quatre appels de commande.

### Procédure

- Pour créer manuellement le document XML de définition de tâche conformément au schéma `FileTransfer.xsd`, voir «Création manuelle d'un fichier XML de définition de tâche en fonction du schéma», à la page 245.
- Pour créer une définition de tâche en modifiant un document généré, éditez le document XML généré par le paramètre **fteCreateTransfer -gt**. Pour plus d'informations, voir «Création d'un document de définition de tâche en modifiant un document généré», à la page 248.

### **Création manuelle d'un fichier XML de définition de tâche en fonction du schéma**

Vous pouvez créer manuellement un fichier XML de définition de tâche en fonction du schéma `FileTransfer.xsd`.

## Pourquoi et quand exécuter cette tâche

Le schéma `FileTransfer.xsd` se trouve dans `MQ_INSTALLATION_PATH/mqft/samples/schema`. Pour plus d'informations sur ce schéma, voir [Format de message de demande de transfert de fichier](#).

### Exemple

L'exemple suivant illustre un exemple de document XML de définition de tâche sauvegardé sous `cleantask.xml`, qui utilise l'élément `<managedCall>` pour appeler un script Ant appelé `RunCleanup.xml`. Le script `RunCleanup.xml` Ant doit se trouver dans le `commandPath` de l'agent de surveillance.

```
<?xml version="1.0" encoding="UTF-8"?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedCall>
    <originator>
      <hostName>hostName</hostName>
      <userID>userID</userID>
      <mqmdUserID>mqmdUserID</mqmdUserID>
    </originator>
    <agent QMgr="QM1" agent="AGENT1"/>
    <reply QMGR="QM1">reply</reply>
    <transferSet priority="1">
      <metaDataSet>
        <metaData key="name1">value1</metaData>
      </metaDataSet>
      <call>
        <command name="RunCleanup.xml" type="antscript" retryCount="2"
          retryWait="30" successRC="0">
          <target>check_exists</target>
          <target>copy_to_archive</target>
          <target>rename_temps</target>
          <target>delete_files</target>
          <property name="trigger.filename" value="{FileName}"/>
          <property name="trigger.path" value="{FilePath}"/>
        </command>
      </call>
    </transferSet>
  </job>
  <name>JOBCLEAN1</name>
</job>
</managedCall>
</request>
```

L'élément `<agent>` spécifie le Managed File Transfer Agent qui est configuré avec le script Ant nommé sur son chemin de commande `commandPath`.

La structure `<call><command>` . . . définit l'exécutable ou le script que vous souhaitez exécuter. La commande prend un attribut `type` facultatif qui peut avoir l'une des valeurs suivantes:

#### antscript

Exécutez un script Ant dans une machine virtuelle Java distincte.

#### exécutable

Appelez un programme exécutable.

#### JCL

Appelez un travail JCL.

Si vous omettez l'attribut `type`, la valeur par défaut `executable` est utilisée.

L'attribut `name` indique le nom du script Ant, de l'exécutable ou du travail JCL que vous souhaitez exécuter, sans aucune information de chemin. L'agent recherche le script ou le programme dans les emplacements spécifiés par la propriété `commandPath` dans le fichier `agent.properties` de l'agent.

L'attribut `retrycount` indique le nombre de tentatives d'appel du programme si le programme ne renvoie pas de code retour de réussite. La valeur affectée à cet attribut ne doit pas être négative. Si vous ne spécifiez pas l'attribut `retrycount`, la valeur par défaut zéro est utilisée.

L'attribut `retrywait` indique le délai d'attente, en secondes, avant une nouvelle tentative d'appel du programme. La valeur affectée à cet attribut ne doit pas être négative. Si vous ne spécifiez pas l'attribut `retrywait`, la valeur par défaut zéro est utilisée.

L'attribut `successrc` est une expression utilisée pour déterminer quand l'appel de programme s'exécute correctement. Le code retour du processus pour la commande est évalué à l'aide de cette expression. La valeur peut être composée d'une ou de plusieurs expressions combinées avec un caractère de barre verticale (|) pour signifier la valeur booléenne OR ou une perluète (&) pour indiquer la valeur booléenne AND. Chaque expression peut être l'un des types d'expression suivants:

- Nombre indiquant un test d'égalité entre le code retour du processus et le nombre.
- Nombre précédé d'un caractère supérieur à (>) pour indiquer un test supérieur à entre le nombre et le code retour du processus.
- Nombre précédé d'un caractère inférieur à (<) pour indiquer un test inférieur à entre le nombre et le code retour du processus.
- Nombre précédé d'un point d'exclamation (!) pour indiquer un test non égal à entre le nombre et le code retour du processus. Par exemple: > 2 & < 7 & ! 5 | 0 | 14 est interprété comme les codes retour suivants ayant abouti: 0, 3, 4, 6, 14. Tous les autres codes retour sont interprétés comme n'ayant pas abouti.

Si vous ne spécifiez pas l'attribut `successrc`, la valeur par défaut zéro est utilisée. Cela signifie que la commande est considérée comme ayant abouti si, et seulement si, elle renvoie un code de zéro.

Pour un script Ant, vous spécifiez généralement des éléments `<target>` et `<property>`. Les valeurs de l'élément `<target>` doivent correspondre aux noms cible dans le script Ant.

Pour les programmes exécutables, vous pouvez spécifier des éléments `<argument>`. Les éléments d'argument imbriqués spécifient les arguments à transmettre au programme appelé dans le cadre de l'appel de programme. Les arguments de programme sont générés à partir des valeurs spécifiées par les éléments d'argument dans l'ordre dans lequel les éléments d'argument sont détectés. Vous pouvez spécifier zéro ou plusieurs éléments d'argument en tant qu'éléments imbriqués d'un appel de programme.

L'administrateur définit et démarre le moniteur comme normal à l'aide du document XML de définition de tâche qui inclut l'élément `<managedCall>`. Exemple :

```
fteCreateMonitor -ma AGENT1 -mm QM1 -md /monitored -mn MONITOR01 -mt
/tasks/cleanuptask.xml -pi 30 -pu seconds -tr match,*go
```

Le chemin d'accès au document XML de définition de transfert doit se trouver sur le système de fichiers local à partir duquel vous exécutez la commande **fteCreateMonitor** (dans cet exemple, /tasks/cleanuptask.xml). Le document `cleanuptask.xml` est utilisé pour créer le moniteur de ressources uniquement. Toutes les tâches référencées par le document `cleanuptask.xml` (scripts Ant ou travaux JCL) doivent se trouver dans le chemin de commande de l'agent de surveillance. Lorsque la condition de déclenchement du moniteur est satisfaite, toutes les variables du fichier XML de définition de tâche sont remplacées par les valeurs réelles du moniteur. Par exemple, `${FilePath}` est remplacé dans le message de demande envoyé à l'agent avec `/monitored/cleanup.go`. Le message de demande est placé dans la file d'attente de commandes de l'agent. Le processeur de commandes détecte que la demande concerne un appel de programme et démarre le programme indiqué. Si une commande de type `antscript` est appelée, une nouvelle machine virtuelle Java est démarrée et la tâche Ant s'exécute sous la nouvelle machine virtuelle Java. Pour plus d'informations sur l'utilisation de la substitution de variable, voir [Personnalisation des tâches avec la substitution de variable](#).

### Concepts associés

«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

## Référence associée

[Format de message de demande de transfert de fichier](#)

[Propriété commandPath MFT](#)

## Création d'un document de définition de tâche en modifiant un document généré

Vous pouvez créer le document de définition de tâche de contrôle en modifiant le document XML généré par l'option **-gt** de **fteCreateTransfer**.

## Pourquoi et quand exécuter cette tâche

Le document généré comporte un élément `<request>` suivi de `<managedTransfer>`. Pour convertir cette définition de tâche en structure `<managedCall>` valide, procédez comme suit:

## Procédure

1. Remplacez les balises de début et de fin `<managedTransfer>` par des balises `<managedCall>`.
2. Supprimez tout élément `<schedule>` et noeud enfant.
3. Remplacez les balises de début et de fin `<sourceAgent>` par `<agent>` pour qu'elles correspondent aux détails de configuration de l'agent de surveillance.
4. Supprimez les éléments `<destinationAgent>` et `<trigger>`.
5. Supprimez les éléments `<item>`.
6. Supprimez tous les éléments `preSourceCall`, `postSourceCall`, `preDestinationCall` et `postDestinationCall`.
7. Insérez une nouvelle structure `<call>...</call>` dans l'élément `<transferSet>`. Cette structure contient la définition de la commande, comme illustré dans l'exemple suivant:

```
<call>
  <command name="RunCleanup.xml" type="antscript" retryCount="2"
  retryWait="30" successRC="0">
    <target>check_exists</target>
    <target>copy_to_archive</target>
    <target>rename_temps</target>
    <target>delete_files</target>
    <property name="trigger.filename" value="{FileName}"/>
    <property name="trigger.path" value="{FilePath}"/>
  </command>
</call>
```

## Exemple

Vous pouvez également conserver l'élément `<managedTransfer>`, y compris tous les détails de transfert de fichier, et insérer jusqu'à quatre appels de commande. Dans ce cas, vous insérez une sélection des éléments d'appel suivants entre les éléments `<metaDataSet>` et `<item>`:

### Appel preSource

Appelez un programme sur l'agent source avant de démarrer le transfert.

### Appel postSource

Appelez un programme sur l'agent source après avoir terminé le transfert.

### Appel preDestination

Appelez un programme sur l'agent de destination avant de démarrer le transfert.

### Appel postDestination

Appelez un programme sur l'agent de destination une fois le transfert terminé.

Chacun de ces éléments prend la structure d'élément `<command>` comme décrit dans l'exemple précédent. Le schéma `FileTransfer.xsd` définit les types utilisés par les différents éléments d'appel.

L'exemple suivant illustre l'appel `preSource`, l'appel `postSource`, l'appel `preDestination` et l'appel `postDestination` dans un document de définition de tâche:

⋮



```

<transferSet priority="1">
  <metaDataSet>
    <metaData key="key1">value1</metaData>
  </metaDataSet>
  <preSourceCall>
    <command name="send.exe" retryCount="0" retryWait="0" successRC="0"
      type="executable">
      <argument>report1.pdf</argument>
      <argument>true</argument>
    </command>
  </preSourceCall>
  <postSourceCall>
    <command name="//DO_IT.JCL" retryCount="0" retryWait="0" successRC="0"
      type="jcl">
      <argument>argument</argument>
    </command>
  </postSourceCall>
  <preDestinationCall>
    <command name="ant_script.xml" retryCount="0" retryWait="0" successRC="0"
      type="antscript">
      <target>step1</target>
      <property name="name" value="value"/>
    </command>
  </preDestinationCall>
  <postDestinationCall>
    <command name="runit.cmd" retryCount="0" retryWait="0" successRC="0" />
  </postDestinationCall>
  <item checksumMethod="none" mode="binary">
  :

```

Vous pouvez mélanger différents types de commande dans le transfert. Les éléments d'argument, de cible et de propriété sont facultatifs.

## Surveillance d'un répertoire et utilisation de la substitution de variable

Vous pouvez surveiller un répertoire à l'aide de la commande **fteCreateMonitor**. La valeur d'une variable de substitution peut être remplacée dans la définition XML de la tâche et utilisée pour définir le comportement de transfert.

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent source est appelé AGENT\_HOP. Le répertoire surveillé par AGENT\_HOP est appelé /test/monitored. L'agent interroge le répertoire toutes les 5 minutes.

Une fois qu'un fichier .zip est écrit dans le répertoire, l'application qui écrit le fichier dans le répertoire écrit un fichier de déclenchement dans le même répertoire. Le nom du fichier de déclenchement est identique à celui du fichier .zip, mais il possède une extension de fichier différente. Par exemple, une fois le fichier file1.zip écrit dans le répertoire, le fichier file1.go est écrit dans le répertoire. Le moniteur de ressources surveille le répertoire pour les fichiers qui correspondent au modèle \*.go, puis utilise la substitution de variable pour demander un transfert du fichier .zip associé.

### Procédure

1. Créez le code XML de tâche qui définit la tâche que le moniteur exécute lorsqu'il est déclenché.

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/${fileName}{token=1}{separator=.}.zip</file>
        </source>
        <destination type="file" exist="overwrite">

```

```

        <file>/out/${fileName}{token=1}{separator=.}}.zip</file>
    </destination>
</item>
</transferSet>
</managedTransfer>
</request>

```

Les variables qui sont remplacées par les valeurs associées au fichier de déclenchement sont mises en évidence en **gras**. Cette tâche XML est sauvegardée dans le fichier /home/USER1/task.xml

2. Créez un moniteur de ressources pour surveiller le répertoire /test/monitored.

Soumettez la commande suivante :

```

fteCreateMonitor -ma AGENT_HOP -mm QM_HOP -md /test/monitored
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr match,*go -pi 5 -pu minutes

```

3. Un utilisateur ou un programme écrit le fichier jump.zip dans le répertoire /test/monitored, puis écrit le fichier jump.go dans le répertoire.
4. Le moniteur est déclenché par l'existence du fichier jump.go. L'agent substitue les informations relatives au fichier de déclenchement dans le code XML de la tâche.

Cela entraîne la transformation du code XML de la tâche en:

```

<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>blue.example.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_HOP" QMgr="QM_HOP" />
    <destinationAgent agent="AGENT_SKIP" QMgr="QM_SKIP" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <file>/test/monitored/jump.zip</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/out/jump.zip</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>

```

## Résultats

Le transfert défini par la tâche XML est effectué. Le fichier jump.zip est lu dans le répertoire /test/monitored par AGENT\_HOP et transféré dans un fichier appelé /out/jump.zip situé sur le système où AGENT\_SKIP est en cours d'exécution.

### Concepts associés

«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

### Tâches associées

«Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts», à la page 245

Les moniteurs de ressources ne sont pas limités à l'exécution de transferts de fichiers en tant que tâche associée. Vous pouvez également configurer le moniteur pour qu'il appelle d'autres commandes à partir de l'agent de surveillance, notamment des programmes exécutables, des scripts Ant ou des travaux JCL. Pour appeler des commandes, éditez le fichier XML de définition de tâche de contrôle pour inclure un ou plusieurs éléments de commande avec les paramètres d'appel de commande correspondants, tels que les arguments et les propriétés.

### Référence associée

**fteCreateMonitor**: créer un moniteur de ressources MFT

## Exemple: configuration d'une ressource MFT

Vous pouvez spécifier une file d'attente IBM MQ comme ressource à surveiller par un moniteur de ressources à l'aide du paramètre **-mq** avec la commande **fteCreateMonitor**.

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, la ressource à surveiller est la file d'attente *MONITORED\_QUEUE*. Cette file d'attente doit se trouver sur le gestionnaire de files d'attente de l'agent de surveillance, *QM\_NEPTUNE*. La condition pour laquelle la file d'attente est surveillée est la présence d'un groupe complet de messages. La tâche à exécuter si la condition est satisfaite est définie dans le fichier *task.xml*.

**Remarque :** Ne créez pas plus d'un moniteur de ressources pour surveiller une file d'attente individuelle. Si vous le faites, un comportement imprévisible se produit.

### Procédure

Entrez la commande suivante :

```
fteCreateMonitor -ma AGENT_NEPTUNE -mn myMonitor -mm QM_NEPTUNE -mq MONITORED_QUEUE  
-mt task.xml -tr completeGroups -pi 5 -pu minutes
```

Le moniteur vérifie la file d'attente toutes les cinq minutes pour voir si la condition *completeGroups* est vraie. S'il existe un ou plusieurs groupes complets dans la file d'attente, le moniteur exécute la tâche définie dans le fichier *task.xml* une fois pour chaque groupe complet.

### Concepts associés

[«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252](#)

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

### Tâches associées

[«Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts», à la page 245](#)

Les moniteurs de ressources ne sont pas limités à l'exécution de transferts de fichiers en tant que tâche associée. Vous pouvez également configurer le moniteur pour qu'il appelle d'autres commandes à partir de l'agent de surveillance, notamment des programmes exécutables, des scripts Ant ou des travaux JCL. Pour appeler des commandes, éditez le fichier XML de définition de tâche de contrôle pour inclure un ou plusieurs éléments de commande avec les paramètres d'appel de commande correspondants, tels que les arguments et les propriétés.

[«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257](#)

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

## Référence associée

**fteCreateMonitor**: créer un moniteur de ressources MFT

## Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.



**Avertissement** : Les noms de variable ne sont pas sensibles à la casse.

Les variables utilisées pour la substitution ne sont disponibles que pour les conditions de déclenchement positives. Seules les conditions de déclenchement `match` et `fileSize` remplacent les variables. Si une condition `noMatch` est utilisée et qu'il existe des noms de variable de substitution dans la définition de tâche, la tâche n'est pas appelée et le moniteur émet le code retour 110 et le message d'erreur BFGDM0060E.

### Si la ressource contrôlée est une file d'attente

La valeur de toute propriété de message IBM MQ du premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de la tâche.

Les propriétés de message définies par l'utilisateur sont préfixées avec `usr.` mais n'incluent pas ce préfixe dans le nom de la variable. Les noms de variable doivent être précédés d'un signe dollar (\$) et placés entre accolades {}.

Par exemple, `${destFileName}` est remplacé par la valeur de la propriété de message `usr.destFileName` du premier message à lire dans la file d'attente source. Pour plus d'informations, voir les propriétés de message MQ lues par MFT à partir des messages des files d'attente source et «Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257.

Si une variable n'est pas définie en tant que propriété de message, le moniteur signale une erreur BFGDM0060E et renvoie le code retour 110 (échec de la substitution de la variable de la tâche de surveillance). En outre, l'agent écrit le message d'erreur suivant dans son journal des événements (`outputN.log`):

```
BFGDM0113W: Trigger failure for <monitor name> for reason BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Si la consignation du moniteur de ressources modéré ou prolix est activée pour le moniteur, le moniteur écrit le message suivant dans le journal des événements du moniteur de ressources de l'agent (`resmoneventN.log`):

```
BFGDM0060E: A monitor task could not complete as a variable substitution <variable name> was not present.
```

Pour plus d'informations sur la journalisation du moniteur de ressources, voir «Consignation des moniteurs de ressources MFT», à la page 264.

Le tableau suivant indique les variables de substitution fournies par défaut. Par exemple, `${AGENTNAME}` est remplacé par le nom de l'agent du moniteur de ressources.

Tableau 10. Variables de substitution fournies par défaut	
Variable	Description
AgentName	Nom de l'agent du moniteur de ressources.

Tableau 10. Variables de substitution fournies par défaut (suite)

Variable	Description
QUEUENAME	Nom de la file d'attente surveillée.
ENCODING	Codage de caractères du premier message de la file d'attente ou du premier message d'un groupe.
ID message	ID message IBM MQ du premier message de la file d'attente ou du premier message du groupe.
groupID	ID groupe IBM MQ du groupe ou ID message si un seul message est trouvé. Cette variable est uniquement définie si vous contrôlez des groupes complets.
CurrentTimestamp	Horodatage basé sur l'heure locale à laquelle le moniteur a été déclenché. La valeur d'horodatage est unique pour l'agent.
CurrentTimeStamp UTC	Horodatage basé sur l'heure, dans le fuseau horaire UTC, à laquelle le moniteur a été déclenché. La valeur d'horodatage est unique pour l'agent.

### Si la ressource contrôlée est un répertoire

Le tableau suivant présente l'ensemble des noms de variable pouvant être remplacés dans la définition XML de la tâche.

Tableau 11. Variables pouvant être substituées

Variable	Description
FilePath	Nom de chemin complet du fichier de déclenchement.
FileName	Partie du nom de fichier du déclencheur.
Heure de LastModified	Heure de la dernière modification du fichier de déclenchement. Cette heure est exprimée en tant qu'heure locale du fuseau horaire dans lequel l'agent s'exécute et est formatée en tant qu'heure ISO 8601.
Date de LastModified	Date de la dernière modification du fichier de déclenchement. Cette date est exprimée en tant que date locale du fuseau horaire dans lequel l'agent s'exécute et est formatée en tant que date ISO 8601.
LastModifiedTimeUTC	Heure de la dernière modification du fichier de déclenchement. Cette heure est exprimée sous la forme de l'heure locale convertie au fuseau horaire UTC et est formatée sous la forme d'une heure ISO 8601.
LastModifiedDateUTC	Date de la dernière modification du fichier de déclenchement. Cette date est exprimée sous forme de date locale convertie au fuseau horaire TUC et est formatée sous forme de date ISO 8601.
AgentName	Nom de l'agent du moniteur de ressources.
CurrentTimestamp	Horodatage basé sur l'heure locale à laquelle le moniteur a été déclenché. La valeur d'horodatage est unique pour l'agent.
CurrentTimeStampUTC	Horodatage basé sur l'heure dans le fuseau horaire UTC à laquelle le moniteur a été déclenché. La valeur d'horodatage est unique pour l'agent.

### Si la ressource contrôlée est un fichier de déclenchement

Le tableau suivant présente l'ensemble des noms de variable pouvant être remplacés lorsqu'un moniteur de ressources utilise le contenu d'un fichier de déclenchement pour déterminer les fichiers à transférer.

Tableau 12. Variables pouvant être remplacées lors de l'utilisation d'un fichier de déclenchement

Variable	Description
contentSource	Nom de chemin complet du fichier source.
contentDestination	Nom de chemin complet du fichier de destination.

Les noms de variable doivent être précédés d'un signe dollar (\$) et placés entre accolades, { }. Par exemple, `${FilePath}` est remplacé par le chemin de fichier qualifié complet du fichier de déclenchement correspondant.

Il existe deux mots clés spéciaux qui peuvent être appliqués aux noms de variable pour fournir des améliorations supplémentaires. Il s'agit des fonctions suivantes :

#### jeton

Index de jeton à remplacer (à partir de 1 à partir de la gauche et à partir de -1 à partir de la droite)

#### séparateur

Caractère unique permettant de segmenter la valeur de la variable. La valeur par défaut est la barre oblique (/) sur les plateformes AIX and Linux ou la barre oblique inversée (\) sur les plateformes Windows, mais le séparateur peut être tout caractère valide pouvant apparaître dans la valeur de la variable.

Si le mot clé de séparation est indiqué dans un nom de variable, la valeur de la variable est divisée en unités lexicales en fonction du caractère de séparation.

La valeur affectée au mot clé token est utilisée comme index pour sélectionner le jeton à utiliser pour remplacer le nom de la variable. L'index de jeton est relatif au premier caractère de la variable et commence à 1. Si le mot clé token n'est pas spécifié, la totalité de la variable est insérée.

Toutes les valeurs qui sont substituées dans un nom d'agent dans le message XML sont traitées de manière non sensible à la casse. Tous les noms Managed File Transfer Agent sont en majuscules. Si la valeur Paris est remplacée par un attribut d'agent dans le message XML, cette valeur est interprétée comme une référence à l'agent PARIS.

#### Concepts associés

«Exemples: substitution de variable pour les définitions de moniteur de ressources», à la page 254  
Exemples de substitution de variables pour les définitions de moniteur de ressources à l'aide de XML et de IBM MQ Explorer.

#### Tâches associées

Que faire si la substitution de variables entraîne l'accès de plusieurs fichiers à un même nom de fichier

## Exemples: substitution de variable pour les définitions de moniteur de ressources

Exemples de substitution de variables pour les définitions de moniteur de ressources à l'aide de XML et de IBM MQ Explorer.

### Exemples illustrant le fonctionnement de la substitution de variable

En supposant que le chemin d'accès au fichier de déclenchement correspondant est `c:\MONITOR\REPORTS\Paris\Report2009.doc` sur les plateformes Windows et `/MONITOR/REPORTS/Paris/Report2009.doc` sur les plateformes AIX and Linux, les variables sont remplacées comme indiqué dans le tableau suivant.

Spécification de variable	Après la substitution de variable
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=1}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=2}{separator=.}}</code>	Windows: doc AIX and Linux : doc
<code>\${FilePath{token=3}}</code>	Windows : RAPPORTS AIX and Linux : Paris

Vous pouvez également spécifier un index de jeton négatif pour sélectionner des jetons par rapport au dernier caractère de la variable, comme indiqué dans le tableau suivant. Les exemples du tableau utilisent la même valeur de variable, c:\MONITOR\REPORTS\Paris\Report2009.doc sous Windows et /MONITOR/REPORTS/Paris/Report2009.doc sous AIX and Linux.

Spécification de variable	Après la substitution de variable
<code>\${FilePath}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009.doc AIX and Linux : /MONITOR/REPORTS/Paris/Report2009.doc
<code>\${FilePath{token=-2}{separator=.}}</code>	Windows :c:\MONITOR\REPORTS\Paris\Report2009 AIX and Linux : /MONITOR/REPORTS/Paris/Report2009
<code>\${FilePath{token=-2}{separator=\}}</code>	Windows : Paris AIX and Linux : Paris
<code>\${FilePath{token=-4}}</code>	Windows : Monitor AIX and Linux : Monitor

Les variables utilisées pour la substitution ne sont disponibles que pour les conditions de déclenchement positives suivantes et l'option `noSizeChange`, qui est une exception à la règle de condition de déclenchement positive:

- `match`
- `fileSize`
- `noSize-Modification`

Si une condition `noMatch` est utilisée et qu'il existe des noms de variable de substitution dans la définition de tâche, la tâche n'est pas appelée et le moniteur émet le code retour 110 et le message d'erreur BFGDM0060E.

## Exemple d'utilisation de XML

L'exemple suivant de définition de tâche XML utilise le nom de l'agent de surveillance comme agent source pour le transfert (Paris), utilise l'avant-dernier nom de répertoire dans le chemin de fichier comme nom de l'agent de destination pour le transfert (Report2009) et renomme le fichier transféré pour qu'il soit la racine du nom du fichier de déclenchement avec l'extension .rpt.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="{AgentName}" QMgr="QM1" />
    <destinationAgent agent="{FilePath{token=-2}}" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/${FileName{token=1}}{separator=}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Cela entraîne la transformation du code XML de la tâche en:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00" xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT1" QMgr="QM1" />
    <destinationAgent agent="Paris" QMgr="QMD" />
    <transferSet>
      <item mode="binary" checksumMethod="MD5">
        <source recursive="false" disposition="leave">
          <file>c:/incoming/reports/summary/report.doc</file>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/Report2009.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

La variable `{FilePath{token=-2}}` dans l'attribut agent de l'élément `<destinationAgent>` est remplacée par la valeur Paris. Cette valeur est traitée de manière non sensible à la casse et interprétée comme une référence à l'agent PARIS.

## Exemples d'utilisation de IBM MQ Explorer

Lors de la création d'un moniteur de ressources via IBM MQ Explorer, et une fois que les propriétés du moniteur et les conditions de déclenchement ont été spécifiées, l'option permet d'ajouter des éléments de transfert au moniteur. Les exemples suivants montrent comment les variables `{FilePath}` et `{FileName}` peuvent être utilisées dans le **panneau Ajouter un élément de transfert** pour personnaliser les transferts résultant d'une correspondance de moniteur de ressources.

### Exemple 1

Afin de simplement transférer le fichier source à un autre emplacement lorsqu'une condition de déclenchement est remplie, la variable `{FilePath}` peut être utilisée:



- Définissez le **nom de fichier** source sur `${FilePath}`.
- Dans le menu déroulant **Type** de la destination, sélectionnez **Annuaire**.
- Définissez le **nom de fichier** de destination comme emplacement vers lequel vous souhaitez que le fichier source soit transféré. Par exemple, il peut s'agir de `C:\MFT\out\`.

## Exemple 2

Pour transférer le fichier source vers un autre emplacement et modifier l'extension du fichier, la variable `${FileName}` peut être utilisée avec la variable `${FilePath}` :

Dans l'exemple suivant, on suppose que le chemin du fichier source est égal à `C:\MONITOR\REPORTS\Paris\Report2009.doc`:

- Définissez le **nom de fichier** source sur `${FilePath}`.
- Définissez le **nom de fichier** de destination comme emplacement vers lequel vous souhaitez que le fichier source soit transféré, suivi de `${FileName}{token=1}{separator=.}`, suivi de la nouvelle extension du fichier. Par exemple, il peut s'agir de `C:\MFT\out\${FileName}{token=1}{separator=.}.rpt`, qui correspond à `C:\MFT\out\Report2009.rpt` avec le nom du fichier source.

## Exemple 2

Afin d'utiliser une partie du chemin du fichier source pour déterminer la destination du transfert, la variable `${FilePath}` peut être utilisée avec des spécifications de jeton et de séparateur.

Dans l'exemple suivant, on suppose que le chemin du fichier source est égal à `C:\MONITOR\REPORTS\Paris\Report2009.doc`.

Il est possible d'utiliser une partie du chemin du fichier source pour déterminer la destination du fichier. A l'aide de l'exemple de chemin de fichier `C:\MONITOR\REPORTS\Paris\Report2009.doc`, si le fichier devait être transféré dans un dossier en fonction de l'emplacement du fichier source, c'est-à-dire `Paris` dans cet exemple, les opérations suivantes peuvent être effectuées:

- Définissez le **nom de fichier** source sur `${FilePath}`.
- Définissez le **Nom de fichier** de destination comme destination dans laquelle se trouvent les dossiers de chaque emplacement, puis ajoutez la partie de destination du chemin de fichier et le nom de fichier. Par exemple, il peut s'agir de `C:\MFT\out\${FilePath}{token=-2}{separator=\}\${FileName}`, qui correspond à `C:\MFT\out\Paris\Report2009.doc` avec le nom du fichier source.

## Concepts associés

[«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252](#)

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

## Tâches associées

[Que faire si la substitution de variables entraîne l'accès de plusieurs fichiers à un même nom de fichier](#)

## Surveillance d'une file d'attente et utilisation de la substitution de variable

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

## Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent source est appelé AGENT\_VENUS, qui se connecte à QM\_VENUS. La file d'attente surveillée par AGENT\_VENUS est appelée START\_QUEUE et se trouve sur QM\_VENUS. L'agent interroge la file d'attente toutes les 30 minutes.

Lorsqu'un groupe complet de messages est écrit dans la file d'attente, la tâche de surveillance envoie le groupe de messages dans un fichier à l'un des agents de destination, qui se connectent tous au gestionnaire de files d'attente QM\_MARS. Le nom du fichier vers lequel le groupe de messages est transféré est défini par la IBM MQ propriété de message `usr.fileName` sur le premier message du groupe. Le nom de l'agent auquel le groupe de messages est envoyé est défini par la IBM MQ propriété de message `usr.toAgent` sur le premier message du groupe. Si l'en-tête `usr.toAgent` n'est pas défini, la valeur par défaut à utiliser pour l'agent de destination est AGENT\_MAGENTA.

Lorsque vous spécifiez `useGroups="true"`, si vous ne spécifiez pas également `groupId="{GROUPID}"`, le transfert prend simplement le premier message de la file d'attente. Par exemple, si vous utilisez la substitution de variable pour générer le `fileName`, il est possible que le contenu de `a.txt` ne soit pas correct. En effet, `fileName` est généré par le moniteur, mais le transfert reçoit en fait un message qui n'est pas celui qui doit générer le fichier appelé `fileName`.

## Procédure

1. Créez le code XML de tâche qui définit la tâche que le moniteur exécute lorsqu'il est déclenché.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

Les variables qui sont remplacées par les valeurs des en-têtes de message IBM MQ sont mises en évidence en **gras**. Cette tâche XML est sauvegardée dans le fichier `/home/USER1/task.xml`

2. Créez un moniteur de ressources pour surveiller la file d'attente START\_QUEUE.

Soumettez la commande suivante :

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Un utilisateur ou un programme écrit un groupe de messages dans la file d'attente START\_QUEUE. Les propriétés de message IBM MQ suivantes sont définies pour le premier message de ce groupe:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. Le moniteur est déclenché lorsque le groupe complet est écrit. L'agent remplace les propriétés de message IBM MQ dans le code XML de la tâche.

Cela entraîne la transformation du code XML de la tâche en:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

## Résultats

Le transfert défini par la tâche XML est effectué. Le groupe complet de messages lus à partir de la file d'attente START\_QUEUE par AGENT\_VENUS est écrit dans un fichier appelé /reports/larmer.rpt sur le système où s'exécute AGENT\_VIOLET.

## Que faire ensuite

### Transfert de chaque message dans un fichier distinct

Si vous souhaitez surveiller une file d'attente et que chaque message soit transféré dans un fichier distinct, vous pouvez utiliser une technique similaire à celle décrite précédemment dans cette rubrique.

1. Créez le moniteur comme décrit précédemment, en spécifiant le paramètre **-tr completeGroups** dans la commande **fteCreateMonitor**.
2. Dans le code XML de la tâche, indiquez ce qui suit:

```
<queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
```

Toutefois, lorsque vous placez les messages dans la file d'attente source, ne les placez pas dans un groupe IBM MQ. Ajoutez des propriétés de message IBM MQ à chaque message. Par exemple, spécifiez la propriété `usr.filename` avec une valeur de nom de fichier unique pour chaque message. Ainsi, Managed File Transfer Agent traite chaque message de la file d'attente source comme un groupe distinct.

### Concepts associés

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

[«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252](#)

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement

sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

Que faire si les fichiers de destination créés par un transfert démarré par un moniteur de ressources de file d'attente contiennent des données incorrectes

### Tâches associées

«Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts», à la page 245

Les moniteurs de ressources ne sont pas limités à l'exécution de transferts de fichiers en tant que tâche associée. Vous pouvez également configurer le moniteur pour qu'il appelle d'autres commandes à partir de l'agent de surveillance, notamment des programmes exécutables, des scripts Ant ou des travaux JCL. Pour appeler des commandes, éditez le fichier XML de définition de tâche de contrôle pour inclure un ou plusieurs éléments de commande avec les paramètres d'appel de commande correspondants, tels que les arguments et les propriétés.

«Exemple: configuration d'une ressource MFT», à la page 251

Vous pouvez spécifier une file d'attente IBM MQ comme ressource à surveiller par un moniteur de ressources à l'aide du paramètre **-mq** avec la commande **fteCreateMonitor**.

### Référence associée

**fteCreateMonitor**: créer un moniteur de ressources MFT

Propriétés de message MQ lues par MFT à partir des messages des files d'attente source

## Configuration du comportement de relance du moniteur pour les transferts de messages vers des fichiers

Si un transfert de message à fichier déclenché par un moniteur de ressources échoue et quitte le groupe de messages qui a déclenché le moniteur dans la file d'attente, ce transfert est soumis à nouveau à des intervalles d'interrogation ultérieurs. Le nombre de fois où le transfert est soumis à nouveau est limité par la propriété **monitorGroupRetryLimit** de l'agent de surveillance.

### Pourquoi et quand exécuter cette tâche

Chaque fois qu'un nouveau transfert de message à fichier est déclenché, un nouvel ID de transfert est généré pour la tâche de transfert.

Si l'agent est redémarré, le moniteur déclenche à nouveau un transfert même si le nombre de fois où le transfert a été déclenché a dépassé la valeur de **monitorGroupRetryLimit** dans le fichier agent.properties. La valeur de la propriété **monitorGroupRetryLimit** correspond au nombre maximal de fois où un moniteur déclenche à nouveau un transfert de message à fichier si le groupe de messages existe toujours dans la file d'attente. La valeur par défaut de cette propriété est 10. La valeur de cette propriété peut être définie sur n'importe quelle valeur entière positive ou sur -1. Si la valeur -1 est spécifiée pour cette propriété, le moniteur déclenche à nouveau le transfert un nombre illimité de fois, jusqu'à ce que la condition de déclenchement ne soit pas satisfaite.

Si une tentative de transfert entraîne le nombre de fois où le transfert a été déclenché et dépasse la valeur de **monitorGroupRetryLimit**, l'agent écrit une erreur dans son journal des événements.

Un message unique est traité comme s'il s'agissait d'un groupe unique, et le transfert est déclenché à nouveau à chaque intervalle d'interrogation alors que le message reste dans la file d'attente et que le nombre de fois où le transfert a été déclenché est inférieur à la valeur de **monitorGroupRetryLimit**.

Pour définir la propriété **monitorGroupRetryLimit** sur l'agent de surveillance, procédez comme suit:

### Procédure

1. Arrêtez l'agent de surveillance à l'aide de la commande **fteStopAgent**.
2. Editez le fichier `agent.properties` de l'agent de surveillance pour inclure la ligne suivante:

```
monitorGroupRetryLimit=number_of_retries
```

Le fichier `agent.properties` se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/monitoring_agent_name`.

3. Démarrez l'agent de surveillance à l'aide de la commande **`fteStartAgent`**.

### Tâches associées

«Exemple: configuration d'une ressource MFT», à la page 251

Vous pouvez spécifier une file d'attente IBM MQ comme ressource à surveiller par un moniteur de ressources à l'aide du paramètre **`-mq`** avec la commande **`fteCreateMonitor`**.

## Utilisation d'un fichier de déclenchement

Vous pouvez utiliser le contenu d'un fichier de déclenchement dans un moniteur de ressources pour définir un ensemble de fichiers à transférer dans une demande de transfert unique. Chaque fois qu'un fichier de déclenchement correspondant est détecté, son contenu est analysé pour les chemins d'accès aux fichiers source et éventuellement pour les chemins d'accès aux fichiers de destination. Ces chemins de fichier sont ensuite utilisés pour définir des éléments de fichier dans le fichier XML de transfert de tâche que vous spécifiez, qui est soumis en tant que demande de transfert unique à l'agent. La définition du moniteur de ressources détermine si le contenu du déclencheur est activé.

Vous pouvez activer le déclenchement de contenu de fichier lorsque vous créez un moniteur en spécifiant le paramètre **`-tc`** (contenu de déclencheur). Ce paramètre **`-tc`** s'applique uniquement aux options de déclencheur de fichier `match` et `noSizeChange`. Pour plus d'informations sur la création d'un moniteur, voir **`fteCreateMonitor`**: Création d'un moniteur de ressources MFT.

Lors de l'utilisation d'un fichier de contenu de déclencheur, le format par défaut de chaque ligne est:

- Un chemin de fichier source unique, ou
- Un chemin d'accès au fichier source et un chemin d'accès au fichier de destination, séparés par une virgule

où les espaces blancs sont traités dans le cadre des chemins d'accès aux fichiers. Il est possible de modifier le format de ligne par défaut en spécifiant les paramètres **`-tcr`** et **`-tcc`** dans la commande **`fteCreateMonitor`**. Pour plus d'informations, voir «Options avancées», à la page 262.

Une fois qu'un fichier de déclenchement est analysé, une liste de chemins de fichier est générée et appliquée au fichier XML de la tâche de transfert que vous avez spécifié. Comme pour tous les moniteurs, le format XML de la tâche de transfert est un XML de tâche de transfert complet généré par la commande **`fteCreateTransfer`** avec un seul élément ou fichier défini. L'élément unique doit utiliser les variables de substitution `${contentSource}`, et éventuellement `${contentDestination}`, comme remplacements pour les chemins de fichier source et de destination. Le moniteur développe le fichier XML de la tâche de transfert pour inclure un élément de fichier pour chaque ligne (chemin de fichier) dans le fichier de déclenchement.

Vous ne pouvez pas utiliser le déclenchement de contenu de fichier avec le paramètre **`-bs`** car le paramètre **`-tc`** implique une demande de transfert pour chaque fichier de déclenchement.

## Exemple

L'exemple suivant définit un moniteur à déclencher sur un fichier qui se termine par `trig` et lit les chemins d'accès à ce fichier.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

La commande **`fteCreateTransfer`** crée un fichier appelé `task.xml` pour un fichier unique avec le chemin d'accès au fichier source `${contentSource}`. Exemple :

```
<item checksumMethod="MD5" mode="binary">
  <source disposition="leave" recursive="false">
    <file>${contentSource}</file>
```

```
</source>
</item>
```

La commande **fteCreateMonitor** recherche les fichiers qui se terminent par `trig` dans le répertoire `/home/trigdir` et utilise le contenu pour créer une demande de transfert unique basée sur `task.xml` pour tous les chemins de ce fichier de déclenchement. Le format du fichier de déclenchement doit être un chemin de fichier (source uniquement) sur chaque ligne sans séparateur de virgule. Exemple :

```
/home/file/first.txt
/home/file/second.txt
/home/different/third.txt
:
```

Tous les fichiers sont distribués dans le répertoire `/file/destdir` avec son nom de fichier et non son chemin d'accès, c'est-à-dire que `/home/file/first.txt` est distribué à `/file/destdir/first.txt`.

Sinon, si vous remplacez le paramètre **-dd /file/destdir** dans la commande **fteCreateTransfer** par **-df \${contentDestination}** et le format du contenu d'un fichier de déclenchement par *chemin du fichier source, chemin du fichier de destination*, vous pouvez définir des chemins de destination différents pour le même agent de destination. Exemple :

```
/home/file/first.txt,/home/other/sixth.txt
```

L'emplacement de destination devient alors `/home/other/sixth.txt`.

Les variables de substitution peuvent être segmentées. Par exemple, vous pouvez séparer la partie nom de fichier du chemin fourni à l'aide de `${contentDestination{token=-1}}`. Par conséquent, si la destination **fteCreateTransfer** est définie comme **-df /file/destdir/\${contentDestination{token=-1}}**, la nouvelle destination pour `/home/file/first.txt` est `/file/destdir/sixth.txt`.

## Options avancées

Vous pouvez modifier le format de ligne par défaut pour le contenu du fichier de déclenchement à l'aide du paramètre **-tcr regex**. Indiquez une expression régulière qui correspond au format de ligne requis et fournit un ou deux groupes de capture. Le premier groupe de capture est la source et le second, facultatif, le groupe de capture est la destination. Exemple :

- Les chemins source et de destination sont séparés par un trait d'union:

```
((?:[^-]+)-((?:[^-]+)+)
```

Dans cet exemple, le séparateur est défini dans trois emplacements et les trois instances du trait d'union, `-`, peuvent être remplacées par n'importe quel caractère. Veillez à mettre en échappement les caractères spéciaux.

- Les chemins source et de destination sont séparés par une virgule avec des espaces de fin. Les commentaires indiqués par un signe dièse (`#`) sont ignorés.

```
((?:[^\, ]+),((?:[^\, ]+)+) *(?:#.*)+
```

Les chemins de fichier ne peuvent pas contenir le signe dièse (`#`). En règle générale, une entrée se présente comme suit: `/home/source/from.txt,/home/destination/to.txt # some comment`.

Si vous utilisez le paramètre **-tcr**, assurez-vous que l'expression régulière est bien conçue et testée de sorte que l'expression puisse détecter les erreurs et analyser correctement les fichiers de déclenchement.

Vous pouvez inverser l'ordre de la capture à l'aide du paramètre **-tcc destSrc** . Si vous spécifiez ce paramètre, le premier groupe de capture est le chemin d'accès au fichier de destination et le deuxième groupe est le chemin d'accès au fichier source.

## Comment les erreurs sont traitées

### Fichier de déclenchement vide

Si le fichier de déclenchement est vide, le résultat n'est pas un transfert de fichier. Autrement dit, le moniteur crée une demande de transfert mais aucun élément de fichier n'est spécifié.

### Fichier de déclenchement comportant des erreurs

Si une entrée d'un fichier de déclenchement ne parvient pas à être analysée par rapport au format attendu, aucune demande de transfert n'est générée. Un journal des erreurs du moniteur est publié et l'erreur est également consignée dans le journal des événements. Le fichier de déclenchement est marqué comme étant traité et le moniteur ne tente pas de le traiter à nouveau tant que le fichier n'a pas été mis à jour.

### XML de tâche de transfert non concordant

Le code XML de la tâche de transfert doit correspondre au fichier de déclenchement, c'est-à-dire que si le code XML de la tâche de transfert contient à la fois `${contentSource}` et `${contentDestination}`, tous les fichiers de déclenchement de ce moniteur doivent avoir des chemins d'accès aux fichiers source et de destination et inversement. Dans le premier cas, le moniteur signale un échec de substitution de `${contentDestination}` si le fichier de déclenchement fournit uniquement le chemin du fichier source.

## Exemples

L'exemple suivant est un déclencheur de contenu de base dans lequel le contenu d'un fichier de déclencheur comporte uniquement un chemin d'accès au fichier source:

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -dd /file/destdir ${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
```

Le paramètre **-tcr** définit deux groupes de capture d'une séquence de caractères séparés par un espace. Le paramètre et l'option **-tcc destSrc** indiquent que les groupes de capture doivent être traités en tant que destination puis source.

```
fteCreateTransfer -gt task.xml -sa SrcAgent -da DestAgent -df ${contentDestination} $
${contentSource}
fteCreateMonitor -mn TrigMonitor -md /home/trigdir -mt task.xml -ma SrcAgent -tr "match,*.trig"
-tc
-tcr "((?:[^\ ])+) ((?:[^\ ])+)" -tcc destSrc
```

## Concepts associés

«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

## Tâches associées

«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor** . La valeur de toute propriété de message IBM MQ

dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

### Référence associée

**fteCreateMonitor**: créer un moniteur de ressources MFT

**fteCreateTransfer**: démarrer un nouveau transfert de fichier

## Consignation des moniteurs de ressources MFT

Vous pouvez obtenir des informations de diagnostic sur les moniteurs de ressources à l'aide de la journalisation.

### Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser la consignation pour les moniteurs de ressources à l'aide de la commande **fteSetAgentLogLevel** ou du fichier `agent.properties` pour contrôler la consignation du moniteur de ressources.

Notez que les points de trace existants sont toujours utilisés pour capturer des informations.

Les journaux du moniteur de ressources sont écrits dans un fichier nommé `resmoneventN.log`, où *N* représente un nombre ; par exemple, `resmonevent0.log`. Les fichiers journaux des événements enregistrent plusieurs actions qui ont lieu lorsqu'un moniteur interroge une ressource, par exemple un répertoire ou une file d'attente.



**Avertissement** : Tous les moniteurs de ressources d'un agent écrivent dans le même fichier journal.

Pour obtenir un exemple de sortie d'un fichier `resmoneventN.log`, voir [Que faire si votre moniteur de ressources de répertoire MFT ne déclenche pas de fichiers](#).

Le tableau suivant répertorie les types d'événements que le moniteur de ressources écrit dans le fichier journal. La troisième colonne décrit le niveau de journalisation nécessaire pour capturer chaque événement où le niveau le plus bas est INFO et le niveau le plus élevé est VERBOSE.

Notez que la définition d'un niveau de journalisation supérieur permet également d'écrire des événements de niveau inférieur. Par exemple, la définition du niveau de journalisation sur MODÉRÉ permet également d'écrire des événements de niveau INFO, mais pas des événements de niveau VERBOSE.

Numéro	Événement	Niveau de journalisation	Description
1	Moniteur créé	INFORMATION	Un moniteur de ressources a été créé.
2	Moniteur supprimé	INFORMATION	Un moniteur de ressources a été supprimé.
3	Moniteur arrêté	INFORMATION	Un moniteur de ressources a été arrêté.
4	Moniteur démarré	INFORMATION	Un moniteur de ressources a été démarré.
5	Démarrer l'interrogation	INFORMATION	Un moniteur de ressources a démarré un nouveau cycle d'interrogation.
6	Fin de l'interrogation	INFORMATION	Un cycle d'interrogation du moniteur de ressources s'est arrêté.
7	Correspondance de motif	PROLIXE	Un fichier dans le répertoire du moniteur de déclenchement ou un message dans une file d'attente correspondant au modèle spécifié a été trouvé.
8	Non-concordance de modèle	PROLIXE	Un fichier non concordant dans le répertoire du moniteur de déclenchement ou un



Numéro	Événement	Niveau de journalisation	Description
			message dans une file d'attente qui ne correspond pas au modèle spécifié a été trouvé.
9	Demande de transfert	INFORMATION	Un transfert a été lancé par le moniteur de ressources.
10	Répertoire trop profond	PROLIXE	Le répertoire contrôlé par le moniteur de ressources contient plus de sous-répertoires à interroger que le nombre indiqué dans la configuration du moniteur de ressources.
11	Fichier verrouillé	MODÉRÉ	Le fichier de déclenchement surveillé par le moniteur de ressources est verrouillé par un autre processus.
12	Taille de fichier réduite	MODÉRÉ	Le fichier de déclenchement est inférieur à la taille spécifiée dans la configuration du moniteur de ressources.
13	Taille de fichier instable	MODÉRÉ	Le fichier de déclenchement est modifié plus fréquemment que prévu par la configuration du moniteur de ressources.
14	Trop d'interrogations	MODÉRÉ	Un moniteur de ressources a interrogé un fichier de déclenchement instable trop de fois.
15	Éléments mis en correspondance	INFORMATION	Nombre total de fichiers de déclenchement trouvés dans le répertoire interrogé par un moniteur de ressources.
16	Transférer des éléments	INFORMATION	Nombre total d'articles dans la demande de transfert.
17	FDC généré	MODÉRÉ	Un moniteur de ressources a généré une exception.
18	Demande de transfert	INFORMATION	Demande de transfert soumise par le moniteur de ressources.
19	Echec du démarrage du moniteur	MODÉRÉ	Le démarrage d'un moniteur de ressources a échoué.
20	Historique effacé	INFORMATION	Les informations de l'historique du moniteur ont été effacées.
21	Echec de la suppression de l'historique de surveillance	INFORMATION	La tentative d'effacement des informations de l'historique du moniteur a échoué.
22	ID de transfert	INFORMATION	L'ID de la demande de transfert a été soumis par le moniteur.
23	Traitement par lots	INFORMATION	Nombre total de demandes de transfert pour les éléments mis en correspondance: $N$ , où $N$ est un nombre.
V 9.4.0 24	Connecté	PROLIXE	Un moniteur de ressources s'est connecté au gestionnaire de files d'attente d'agent.

Numéro	Événement	Niveau de journalisation	Description
V 9.4.0 25	Déconnecté	PROLIXE	Un moniteur de ressources s'est déconnecté du gestionnaire de files d'attente de l'agent.
V 9.4.0 26	Erreur lors de la déconnexion	PROLIXE	Un moniteur de ressources a rencontré un problème lors de la déconnexion du gestionnaire de files d'attente de l'agent.

## Procédure

- Pour utiliser la commande **fteSetAgentLogLevel** afin d'activer et de désactiver la journalisation du moniteur de ressources, voir [fteSetAgentLogLevel](#) pour obtenir une description du paramètre **logMonitor** et des exemples d'utilisation des différentes options.
- Pour utiliser le fichier `agent.properties` afin de contrôler la consignation du moniteur de ressources, voir [Le fichier MFT agent.properties](#) pour obtenir une description des propriétés supplémentaires qui vous permettent d'effectuer les activités de consignation suivantes:
  - Activer ou désactiver la journalisation
  - Limiter la taille de chaque fichier journal
  - Limitation du nombre de journaux que les moniteurs de ressources peuvent générer

## Exemple

Les exemples de message suivants définissent la verbose consignation de niveau pour l'agent HA2, sur le gestionnaire de files d'attente MFTDEMO:

```
<?xml version="1.0"?>
<log:log version="6.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xmlns:log="https://www.ibm.com/log">
  <log:originator>
    <log:request>
      <log:hostname>192.168.7.1</log:hostname>
      <log:userid>johndoe</log:userid>
    </log:request>
  </log:originator>
  <log:endpoint agent="HA2" QMgr="MFTDEMO"/>
  <log:logMonitor>MON1="verbose"</log:logMonitor>
</log:log>
```

## Référence associée

Commande [fteSetAgentLogLevel](#)

[Le fichier MFT agent.properties](#)

## Démarrage d'un moniteur de ressources MFT

Depuis la IBM MQ 9.3.0, vous pouvez démarrer des moniteurs de ressources sans avoir à arrêter ou à redémarrer un agent à l'aide de la commande **fteStartMonitor**.

## Avant de commencer

Si la gestion des droits d'accès utilisateur a été activée en définissant l'attribut **authorityChecking** sur `true` dans le fichier `agent.properties`, vous devez disposer des droits `Monitor` ou `Monitor operations` pour démarrer un moniteur de ressources. Pour plus d'informations sur la gestion des droits utilisateur, voir [Restriction des droits utilisateur sur les actions de l'agent MFT](#).

## Pourquoi et quand exécuter cette tâche

Vous pouvez exécuter la commande **fteStartMonitor** à partir de n'importe quel système sur lequel le composant de commandes Managed File Transfer est installé, ce qui signifie que vous pouvez démarrer un moniteur de ressources à partir de n'importe quel emplacement et qu'il n'est pas limité au système sur lequel l'agent propriétaire du moniteur de ressources est en cours d'exécution. Pour plus d'informations sur les paramètres obligatoires et facultatifs de cette commande, voir [fteStartMonitor \(start an MFT resource monitor\)](#).

## Procédure

- Pour connaître l'état d'un agent avant ou après l'exécution de la commande **fteStartMonitor**, utilisez la commande **ftelistMonitors** avec le paramètre **-v**, comme illustré dans l'exemple suivant:

```
ftelistMonitors -ma monitoring_agent_name -v
```

- Pour démarrer un moniteur de ressources dans un agent exécuté sur la même machine, entrez la commande **fteStartMonitor** comme suit:

```
fteStartMonitor -mn monitor_name -ma agent_name
```

- Pour démarrer un moniteur de ressources dans un agent exécuté sur une machine différente, entrez la commande **fteStartMonitor** comme suit:

```
fteStartMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Si le gestionnaire de files d'attente de commandes est également le gestionnaire de files d'attente d'agent de l'agent de surveillance, le paramètre **-mm** est facultatif. Sinon, vous devez spécifier le gestionnaire de files d'attente d'agent avec le paramètre **-mm**.

## Résultats

Si l'agent est en cours d'exécution, le moniteur de ressources est démarré s'il est actuellement arrêté. La commande génère les messages suivants et consigne un événement dans le fichier output0.log de l'agent.

```
BFGCL0816I: Une demande de démarrage du moniteur de ressources 'nom_moniteur' de  
l'agent 'nom_agent' a été émise.  
BFGCL0251I: La demande a abouti.
```

Pour plus d'informations sur les messages générés par la commande s'il est impossible de démarrer le moniteur de ressources, voir [fteStartMonitor \(start an MFT resource monitor\)](#).

### Concepts associés

«[Concepts de surveillance des ressources MFT](#)», à la page 240

Présentation des concepts clés de la fonction de surveillance des ressources Managed File Transfer .

### Tâches associées

«[Arrêt d'un moniteur de ressources MFT](#)», à la page 267

Depuis la IBM MQ 9.3.0, vous pouvez arrêter les moniteurs de ressources sans avoir à arrêter ou à redémarrer un agent à l'aide de la commande **fteStopMonitor** .

### Référence associée

[Moniteur fteStart\(démarrage d'un moniteur de ressources MFT\)](#)

## Arrêt d'un moniteur de ressources MFT

Depuis la IBM MQ 9.3.0, vous pouvez arrêter les moniteurs de ressources sans avoir à arrêter ou à redémarrer un agent à l'aide de la commande **fteStopMonitor** .

## Avant de commencer

Si la gestion des droits utilisateur a été activée en définissant l'attribut **authorityChecking** sur `true` dans le fichier `agent.properties`, vous devez disposer des droits Surveillance ou Opérations de surveillance pour arrêter un moniteur de ressources. Pour plus d'informations sur la gestion des droits utilisateur, voir [Restriction des droits utilisateur sur les actions de l'agent MFT](#).

## Pourquoi et quand exécuter cette tâche

Vous pouvez exécuter la commande **fteStopMonitor** à partir de n'importe quel système sur lequel le composant de commandes Managed File Transfer est installé, ce qui signifie que vous pouvez arrêter un moniteur de ressources à partir de n'importe quel emplacement et qu'il n'est pas limité au système sur lequel l'agent propriétaire du moniteur de ressources est en cours d'exécution. Pour plus d'informations sur les paramètres obligatoires et facultatifs de cette commande, voir [fteStopMonitor \(stop an MFT resource monitor\)](#).

Lorsqu'un moniteur de ressources est arrêté, il écrit un message dans le journal des événements du moniteur de ressources de l'agent, `resmoneventnumber.log`. Si le moniteur de ressources est arrêté à l'aide de la commande **fteStopMonitor**, le message inclut le nom de l'utilisateur qui a émis la demande d'arrêt:

```
Moniteur de ressources arrêté par l'utilisateur '< id_mquser >'
```

Un moniteur de ressources est automatiquement démarré si son agent redémarre, même s'il a été précédemment arrêté à l'aide de la commande **fteStopMonitor**.

Les agents traitent les demandes d'arrêt du moniteur en série plutôt qu'en parallèle. Par exemple, si un agent reçoit une demande d'arrêt du moniteur M1, puis une autre demande d'arrêt du moniteur M2 en succession rapide, il arrête M1 avant de tenter d'arrêter M2.

## Procédure

- Pour connaître l'état d'un agent avant ou après l'exécution de la commande **fteStopMonitor**, utilisez la commande **ftelistMonitors** avec le paramètre **-v**, comme illustré dans l'exemple suivant:

```
ftelistMonitors -ma monitoring_agent_name -v
```

- Pour arrêter un moniteur de ressources dans un agent exécuté sur la même machine, entrez la commande **fteStopMonitor** comme suit:

```
fteStopMonitor -mn monitor_name -ma agent_name
```

- Pour arrêter un moniteur de ressources dans un agent exécuté sur une machine différente, entrez la commande **fteStopMonitor** comme suit:

```
fteStopMonitor -mn monitor_name -ma agent_name -mm AgentQueueManager
```

Si le gestionnaire de files d'attente de commandes est également le gestionnaire de files d'attente d'agent de l'agent de surveillance, le paramètre **-mm** est facultatif. Sinon, vous devez spécifier le gestionnaire de files d'attente d'agent avec le paramètre **-mm**.

## Résultats

Si l'agent est en cours d'exécution, le moniteur de ressources est arrêté s'il est actuellement démarré. La commande génère les messages suivants et consigne un événement dans le fichier `output0.log` de l'agent.

```
BFGCL0813I: Une demande d'arrêt du moniteur de ressources 'MNTR' de l'agent 'SOURCE' a été émise.  
BFGCL0251I: La demande a abouti.
```

Pour plus d'informations sur les messages générés par la commande si elle ne parvient pas à arrêter le moniteur de ressources, voir [fteStopMonitor \(stop an MFT resource monitor\)](#).

## Concepts associés

«Concepts de surveillance des ressources MFT», à la page 240  
Présentation des concepts clés de la fonction de surveillance des ressources Managed File Transfer .

## Tâches associées

«Démarrage d'un moniteur de ressources MFT», à la page 266  
Depuis la IBM MQ 9.3.0, vous pouvez démarrer des moniteurs de ressources sans avoir à arrêter ou à redémarrer un agent à l'aide de la commande **fteStartMonitor** .

## Référence associée

[fteStopMonitor \(arrêt d'un moniteur de ressources MFT\)](#)

## Sauvegarde et restauration des moniteurs de ressources MFT

Vous pouvez sauvegarder les moniteurs de ressources que vous souhaitez pouvoir utiliser ultérieurement en exportant leurs définitions dans un fichier XML que vous pouvez ensuite importer pour créer un nouveau moniteur de ressources à partir de la sauvegarde.

### Pourquoi et quand exécuter cette tâche

Vous devrez peut-être sauvegarder les moniteurs de ressources que vous avez précédemment définis afin de pouvoir réutiliser leurs définitions ultérieurement, par exemple pour recréer les moniteurs de ressources dans une autre infrastructure ou si un moniteur de ressources doit être recréé en raison de problèmes liés au gestionnaire de files d'attente.

Vous pouvez sauvegarder une définition de gestionnaire de ressources unique à l'aide de la commande **fteCreateMonitor** ou de la commande **fteListMonitors** avec le paramètre **-ox** . Dans les deux cas, la définition du gestionnaire de ressources est sauvegardée en l'exportant dans un fichier XML. Vous pouvez ensuite utiliser le paramètre **-ix** de la commande **fteCreateMonitor** pour créer un gestionnaire de ressources en important la définition à partir du fichier XML.

Avec le paramètre **-ox** , vous ne pouvez sauvegarder qu'une seule définition de moniteur de ressources à la fois.

Le paramètre **-od** a été ajouté à la commande **fteListMonitors**. En spécifiant ce paramètre, vous pouvez sauvegarder plusieurs moniteurs de ressources à la fois en exportant leurs définitions en bloc dans un répertoire spécifié. Chaque définition de moniteur de ressources est sauvegardée dans un fichier XML distinct avec un nom au format *agent name.monitor name.xml*.

Le paramètre **-od** est particulièrement utile si vous disposez d'un grand nombre de moniteurs de ressources que vous souhaitez sauvegarder car vous devez exécuter la commande **fteListMonitors -od** une seule fois, au lieu d'exécuter la commande **fteListMonitors -ox** séparément pour chaque définition de ressource, ou utiliser un script distinct pour exécuter la commande **fteListMonitors -ox** pour chaque moniteur de ressources.

## Procédure

- Pour sauvegarder la définition d'un moniteur de ressources en l'exportant dans un fichier XML, utilisez l'une des commandes suivantes:

- La commande **fteCreateMonitor** avec le paramètre **-ox** .
- La commande **fteListMonitors** avec le paramètre **-ox** .

Lorsque vous utilisez le paramètre **-ox** , vous devez également spécifier les paramètres **-ma** et **-mn** , comme illustré dans l'exemple suivant:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -ox filename1.xml
```

- Pour sauvegarder plusieurs définitions de moniteur de ressources en les exportant vers des fichiers XML dans un répertoire spécifié, utilisez la commande **fteListMonitors** avec le paramètre **-od** , comme illustré dans l'exemple suivant:

```
fteListMonitors -od /usr/mft/resmonbackup
```

Vous devez spécifier un répertoire cible valide lorsque vous sauvegardez des moniteurs de ressources en bloc. Si vous ne spécifiez pas de chemin cible, un message d'erreur est généré, comme illustré dans l'exemple suivant:

```
BFGCL0762E: Répertoire de sortie non spécifié. Réexécutez la
commande en spécifiant un chemin valide.
```

Le paramètre **-od** ne doit pas être combiné avec le paramètre **-ox**, sinon le message d'erreur suivant s'affiche:

```
BFGCL0761E: Il n'est pas valide de spécifier à la fois les paramètres '-od' et '-ox'.
```

Vous pouvez définir un ensemble particulier de moniteurs de ressources à inclure dans la sauvegarde. Par exemple, en utilisant le paramètre **-ma** pour spécifier le nom d'un agent, vous pouvez sauvegarder tous les moniteurs de ressources de cet agent, comme illustré dans l'exemple suivant:

```
fteListMonitors -ma AGENT1 -od /usr/mft/resmonbackup
```

Vous pouvez également utiliser la mise en correspondance générique en incluant un astérisque (\*) lors de la définition d'un modèle à utiliser pour la mise en correspondance des noms d'agent ou des noms de moniteur, ou les deux. L'exemple suivant sauvegarde tous les moniteurs de ressources dont les noms correspondent à un modèle spécifié et qui se trouvent dans un agent dont le nom correspond à un modèle spécifié:

```
fteListMonitors -ma AGENT* -mn MON* -od /usr/mft/resmonbackup
```

Lorsque la commande est en cours d'exécution, elle affiche les messages de rapport de progression suivants:

```
Un total de nombre de définitions de moniteur de ressources correspondantes a été trouvé.
index de nombre définitions de moniteur de ressources sauvegardées dans le système de fichiers.
```

Si vous utilisez l'option prolixe, le total en cours d'exécution est toujours affiché, mais au lieu d'afficher *index* de *nombre* définitions de moniteur de ressources sauvegardées dans le système de fichiers la commande affiche le nom de la définition de moniteur en cours de sauvegarde, par exemple:

```
BFGCL0762I: Définition du moniteur 'FILEMON' de l'agent 'XFERAGENT' sauvegardé en tant que
FILEMON.XFERAGENT.XML vers le système de fichiers.
```

- Pour sauvegarder un moniteur de ressources pour un agent particulier en l'exportant dans un fichier XML dans un répertoire spécifié, utilisez la commande **fteListMonitors** avec le paramètre **-od**:

```
fteListMonitors -ma AGENT1 -mn MONITOR1 -od /usr/mft/resmonbackup
```

L'utilisation du paramètre **-od** pour sauvegarder un moniteur de ressources unique est similaire à l'utilisation du paramètre **-ox**, sauf que le nom du fichier de sortie est au format *agent.name.monitor.name.xml*.

- Pour restaurer des définitions de moniteur de ressources à partir d'une sauvegarde, utilisez la commande **fteCreateMonitor** avec le paramètre **-ix**, comme illustré dans l'exemple suivant:

```
fteCreateMonitor -ix file name
```

Pour plus d'exemples d'utilisation du paramètre **-od**, voir [fteListMonitors: list MFT resource monitors](#).

### Référence associée

**fteCreateMonitor**: créer un moniteur de ressources MFT

[fteListMonitors](#): répertorie les moniteurs de ressources MFT

## Effacement de l'historique du moniteur de ressources

Vous pouvez effacer l'historique d'un moniteur de ressources pour qu'une autre demande de transfert de fichier puisse être soumise pour un fichier qui n'a pas été transféré plus tôt en raison d'un

échec. Pour effacer l'historique du moniteur de ressources, vous pouvez utiliser la commande **fteClearMonitorHistory** ou IBM MQ Explorer.

## Avant de commencer

Si la gestion des droits utilisateur a été activée en définissant l'attribut **authorityChecking** sur true dans le fichier `agent.properties`, l'utilisateur qui efface l'historique du moniteur doit disposer des droits appropriés, comme indiqué dans le tableau suivant.

Tableau 15. Droits utilisateur requis pour exécuter la commande **fteClearMonitorHistory**

Utilisateur effaçant l'historique du moniteur	Droits d'accès MFT	Droits requis
Le même utilisateur que celui qui a créé le moniteur de ressources.	Moniteur	BROWSE on SYSTEM.FTE.AUTHMON1. <nom_agent_surveillance>  Il s'agit des mêmes droits que ceux requis pour créer ou supprimer le moniteur de ressources.
Tout utilisateur autre que celui qui a créé le moniteur de ressources.	Monitor operations	SET on SYSTEM.FTE.AUTHPS1. <nom_agent>  Il s'agit des mêmes droits que ceux requis pour supprimer le moniteur de ressources.

Pour plus d'informations sur la gestion des droits utilisateur, voir [Restriction des droits utilisateur sur les actions de l'agent MFT](#).

Si un utilisateur sans les droits requis tente d'effacer l'historique du moniteur de ressources, la commande **fteClearMonitorHistory** génère un message d'erreur et consigne l'échec dans le fichier `output0.log` de l'agent. Pour plus d'informations, voir [fteClearMonitorHistory: clear resource monitor history](#).

## Pourquoi et quand exécuter cette tâche

Si un transfert de fichier a été lancé et qu'un fichier ne peut pas être transféré pour une raison quelconque, le moniteur de ressources ne le sélectionne pas à nouveau pour le transfert lors de l'interrogation suivante car l'historique du moniteur indique que le fichier a été vu lors d'une interrogation antérieure et qu'il n'a pas été modifié depuis (voir «[Concepts de surveillance des ressources MFT](#)», à la page 240).

Avant IBM MQ 9.1.3, si le transfert d'un fichier échoue, le transfert de fichier ne peut être initié à nouveau que si le fichier est supprimé, puis placé à nouveau dans le répertoire, ou si le fichier est mis à jour de sorte que l'attribut de date de dernière modification soit modifié, ou si le moniteur de ressources lui-même est recréé.

Toutefois, vous pouvez effacer l'historique du moniteur de ressources à l'aide de la commande **fteClearMonitorHistory** ou de l'IBM MQ Explorer. L'effacement de l'historique permet à une autre demande de transfert d'un fichier dont le transfert a échoué d'être soumise sans qu'il soit nécessaire de supprimer le fichier, puis de le placer à nouveau dans le répertoire, ou de mettre à jour le fichier pour modifier son attribut de date de dernière modification, ce qui est utile, par exemple, dans les cas où il est nécessaire de transférer le fichier, mais qu'il n'est pas possible de modifier le fichier. De plus, si vous effacez l'historique d'un moniteur de ressources, il n'est pas nécessaire de recréer le moniteur de ressources afin de soumettre une nouvelle demande de transfert d'un fichier dont le transfert a échoué.

L'exemple de membre SCSQFCMD fourni avec Managed File Transfer sous z/OS inclut un script JCL pour effacer l'historique d'un moniteur.

## Procédure

- Pour utiliser la commande **fteClearMonitorHistory** afin d'effacer l'historique du moniteur de ressources, entrez la commande au format suivant:

```
fteClearMonitorHistory -p <configuration> -ma <agent name> -mn <monitor name> -w 1000
```

Seuls les paramètres **-ma** et **-mn** sont requis. Tous les autres paramètres sont facultatifs. Pour plus d'informations sur l'utilisation de la commande **fteClearMonitorHistory**, y compris des exemples, voir [fteClearMonitorHistory: clear resource monitor history](#).

Si l'historique est effacé, la commande génère le message suivant:

```
BFGCL0780I: Une demande d'effacement de l'historique du moniteur de ressources'nom du
moniteur'de l'agent'nom de l'agent'a été émise.
BFGCL0251I: La demande a abouti.
```

et consigne la réussite dans le fichier output0.log de l'agent.

Si la tentative d'effacement de l'historique du moniteur de ressources échoue, **fteClearMonitorHistory** génère un message d'erreur et consigne l'échec dans le fichier output0.log de l'agent.

- Pour utiliser la vue du moniteur de ressources dans le plug-in IBM MQ Explorer MFT afin d'effacer l'historique du moniteur de ressources, cliquez avec le bouton droit de la souris sur le moniteur de ressources et sélectionnez **Effacer l'historique** dans le menu déroulant.

Si l'historique est effacé avec succès, le message suivant s'affiche:

```
BFGUI00171: L'historique du moniteur de ressources a été effacé.
```

Si la tentative d'effacement de l'historique échoue, un message d'erreur s'affiche. Exemple :

```
BFGUI0016E Echec de l'effacement de l'historique du moniteur de ressources spécifié-raison 2059
```

## Utilisation des modèles de transfert de fichiers

Vous pouvez utiliser des modèles de transfert de fichiers pour stocker des paramètres de transfert de fichiers communs pour des transferts répétés ou complexes. Créez un modèle de transfert à partir de la ligne de commande à l'aide de la commande **fteCreateTemplate** ou utilisez la IBM MQ Explorer pour créer un modèle de transfert à l'aide de l'assistant **Créer un modèle pour Managed File Transfer**, ou sauvegardez un modèle lors de la création d'un transfert de fichier en cochant la case **Sauvegarder les paramètres de transfert en tant que modèle**. La fenêtre **Modèles de transfert** affiche tous les modèles de transfert que vous avez créés dans votre réseau Managed File Transfer.

### Pourquoi et quand exécuter cette tâche

Pour créer un modèle de transfert à partir de la ligne de commande, utilisez la commande **fteCreateTemplate**. Ensuite, lorsque vous souhaitez soumettre un modèle de transfert que vous avez créé sur la ligne de commande, cliquez sur **Soumettre** dans IBM MQ Explorer.

Pour afficher les modèles de transfert dans le IBM MQ Explorer, procédez comme suit:


### Procédure

- Développez **Managed File Transfer** dans la vue Navigator. **Managed File Transfer Central** s'affiche dans la vue Contenu.
- Tous les gestionnaires de files d'attente de coordination sont répertoriés dans la vue Navigator. Développez le nom du gestionnaire de file d'attente de coordination que vous avez utilisé pour le transfert planifié. Si vous souhaitez modifier le gestionnaire de file d'attente de coordination auquel vous êtes connecté, cliquez avec le bouton droit de la souris sur le nom du gestionnaire de file d'attente de coordination à utiliser dans la vue Navigator, puis cliquez sur **Connexion**.



3. Cliquez sur **Modèles de transfert**. La fenêtre **Modèles de transfert** s'affiche dans la vue Contenu.
4. La fenêtre **Modèles de transfert** répertorie les détails suivants sur vos transferts de fichiers:
  - a) **Nom** Nom de votre modèle de transfert de fichiers.
  - b) **Source** Nom de l'agent utilisé pour transférer le fichier à partir du système source.
  - c) **Fichier source** Nom du fichier à transférer sur son système hôte.  
Développez les informations du modèle de transfert pour afficher cette zone.
  - d) **Destination** Nom de l'agent utilisé pour recevoir le fichier sur le système de destination.
  - e) **Fichier de destination** Nom du fichier après son transfert vers le système de destination.  
Développez les informations du modèle de transfert pour afficher cette zone.
  - f) **Début planifié (fuseau horaire sélectionné)** Date et heure auxquelles le transfert de fichier est planifié pour démarrer dans le fuseau horaire utilisé par l'administrateur. Pour modifier le fuseau horaire affiché, cliquez sur **Fenêtre > Préférences > IBM MQ Explorer > Managed File Transfer** et sélectionnez un autre fuseau horaire dans la liste **Fuseau horaire**. Cliquez sur **OK**.
  - g) **Événements déclencheurs** Type d'événement qui déclenche le démarrage du transfert de fichiers. Le type peut être l'une des valeurs suivantes: existe, n'existe pas, ou dépasse.

## Résultats

Pour actualiser les éléments affichés dans la fenêtre **Modèles de transfert**, cliquez sur le bouton Actualiser  dans la barre d'outils de la vue Contenu.

Pour soumettre un modèle de transfert et démarrer le transfert défini dans le modèle, cliquez avec le bouton droit de la souris sur le nom du modèle et cliquez sur **Soumettre**.

Pour modifier un modèle de transfert, cliquez avec le bouton droit de la souris sur le nom du modèle et cliquez sur **Editer**. Tous les fichiers inclus dans le modèle d'origine sont répertoriés comme faisant partie d'un groupe de transfert, même s'ils n'ont pas été inclus comme faisant partie d'un groupe dans le modèle d'origine. Si vous souhaitez supprimer un fichier du modèle, vous devez sélectionner la spécification de fichier dans le groupe et cliquer sur **Supprimer la sélection**. Si vous souhaitez ajouter de nouvelles spécifications de fichier au modèle, utilisez les zones du panneau Modèle et cliquez sur le bouton **Ajouter au groupe**. Une fois que vous avez effectué vos modifications, vous êtes invité à attribuer un nouveau nom au modèle édité.

Pour créer un transfert de fichier à partir d'un modèle de transfert, cliquez avec le bouton droit de la souris sur le nom du modèle et cliquez sur **Editer en tant que nouveau transfert**.

Pour créer une copie en double d'un modèle de transfert, cliquez avec le bouton droit de la souris sur le nom du modèle et cliquez sur **Dupliquer**. Le modèle de transfert en double est automatiquement sauvegardé avec le même nom que le modèle d'origine, auquel est ajouté "(copy)".

Pour supprimer un modèle de transfert, cliquez avec le bouton droit de la souris sur le nom du modèle, puis cliquez sur **Supprimer**.

### Tâches associées

«Création d'un modèle de transfert de fichiers à l'aide de IBM MQ Explorer», à la page 274

Vous pouvez créer un modèle de transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande. Vous pouvez ensuite utiliser ce modèle pour créer de nouveaux transferts de fichiers à l'aide des détails du modèle ou soumettre le modèle pour démarrer le transfert de fichiers.

### Référence associée

**[fteCreateTemplate](#)**: créer un nouveau modèle de transfert de fichiers

[fteListTemplates](#)

[fteDeleteTemplates](#)

## Création d'un modèle de transfert de fichiers à l'aide de IBM MQ Explorer

Vous pouvez créer un modèle de transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande. Vous pouvez ensuite utiliser ce modèle pour créer de nouveaux transferts de fichiers à l'aide des détails du modèle ou soumettre le modèle pour démarrer le transfert de fichiers.

### Pourquoi et quand exécuter cette tâche

Pour créer un modèle de transfert de fichier à partir de la ligne de commande, utilisez la commande [fteCreateTemplate](#).

Pour créer un modèle de transfert de fichiers à l'aide de l'assistant **Créer un modèle pour Managed File Transfer** dans IBM MQ Explorer, procédez comme suit:

### Procédure

1. Dans la vue Navigator, cliquez sur **Transfert de fichier géré. Managed File Transfer Central** s'affiche dans la vue Contenu.
2. Tous vos gestionnaires de files d'attente de coordination sont affichés dans la vue Navigator. Développez le nom du gestionnaire de file d'attente de coordination que vous avez utilisé pour le transfert planifié. Si vous souhaitez modifier le gestionnaire de file d'attente de coordination auquel vous êtes connecté, cliquez avec le bouton droit de la souris sur le nom du gestionnaire de file d'attente de coordination à utiliser dans la vue Navigator, puis cliquez sur **Connexion**.
3. Démarrez l'assistant **Créer un modèle pour Managed File Transfer** en cliquant avec le bouton droit de la souris sur **Modèles de transfert**, puis en cliquant sur **Nouveau modèle**.
4. Suivez les instructions des panneaux de l'assistant. Une aide contextuelle est fournie pour chaque panneau. Pour accéder à l'aide contextuelle sur Windows, appuyez sur F1. Sous Linux, appuyez sur Ctrl+F1 ou sur Shift+F1.

Si vous avez créé un modèle contenant tous les détails de transfert requis, veillez à cocher la case **Sauvegarder les paramètres de transfert en tant que modèle** sur la page **Récapitulatif du transfert** si cette case n'est pas déjà cochée. Entrez également un nom pour le modèle dans la zone Nom. Si vous créez un modèle qui ne contient pas encore tous les détails de transfert requis, la case **Sauvegarder les paramètres de transfert en tant que modèle** est automatiquement cochée pour vous.

### Tâches associées

«Utilisation des modèles de transfert de fichiers», à la page 272

Vous pouvez utiliser des modèles de transfert de fichiers pour stocker des paramètres de transfert de fichiers communs pour des transferts répétés ou complexes. Créez un modèle de transfert à partir de la ligne de commande à l'aide de la commande **fteCreateTemplate** ou utilisez la IBM MQ Explorer pour créer un modèle de transfert à l'aide de l'assistant **Créer un modèle pour Managed File Transfer**, ou sauvegardez un modèle lors de la création d'un transfert de fichier en cochant la case **Sauvegarder les paramètres de transfert en tant que modèle**. La fenêtre **Modèles de transfert** affiche tous les modèles de transfert que vous avez créés dans votre réseau Managed File Transfer.

### Référence associée

**fteCreateTemplate**: créer un nouveau modèle de transfert de fichiers

[fteListTemplates](#)

[fteDeleteTemplates](#)

### Sauvegarde d'une définition de modèle de transfert de fichier

Les modèles de transfert de fichiers contiennent un document XML qui définit les spécifications de fichier source et de destination pour le transfert. Vous pouvez utiliser ce fichier XML en entrée de la commande **fteCreateTemplate** pour recréer un modèle de transfert de fichier.

## Pourquoi et quand exécuter cette tâche

Pour sauvegarder le document XML contenant les spécifications de fichier source et de destination pour un modèle de transfert, utilisez la commande `fteCreateTransfer` ou IBM MQ Explorer. Pour créer un fichier de sauvegarde au format XML de modèle de transfert, procédez comme suit:

### Procédure

- Première méthode: utilisez le paramètre **-gt** sur une commande `fteCreateTransfer` pour générer un message XML de modèle de transfert dans un nouveau fichier.
- Méthode deux: créez le modèle à l'aide de IBM MQ Explorer.  
Lorsque vous accédez à la page *Récapitulatif du modèle de transfert* :
  - a) Copiez l' *aperçu XML du message de demande*.
  - b) Sauvegardez ce message XML de modèle de transfert dans un nouveau fichier.
- Méthode trois: utilisez IBM MQ Explorer pour sauvegarder les modèles existants.
  - a) Accédez à **Managed File Transfer > Nom du gestionnaire de files d'attente > Modèles de transfert**.
  - b) Dans le panneau Transfert, mettez en évidence le modèle à sauvegarder, cliquez avec le bouton droit de la souris et sélectionnez **Editer** dans le menu contextuel.
  - c) Cliquez sur **Suivant** jusqu'à la page *Récapitulatif du modèle de transfert* .
  - d) Copiez l' *aperçu XML du message de demande*.
  - e) Sauvegardez ce message XML de modèle de transfert dans un nouveau fichier.

### Résultats

Vous pouvez utiliser le fichier de messages XML de modèle de transfert, créé par l'une des méthodes ci-dessus, comme entrée dans la commande `fteCreateTemplate` . Pour plus d'informations sur l'utilisation de cette commande, voir la commande `fteCreateTemplate` .

#### Référence associée

[Commande fteCreateTemplate](#)

[Commande Tmplates fteListt](#)

## Transfert de données de fichiers vers des messages

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

Pour plus d'informations sur les transferts de messages vers des fichiers, voir [«Transfert de données de messages vers des fichiers»](#), à la page 291.

L'agent de destination pour un transfert de fichier à message ne peut pas être un agent de pont de protocole ou un agent de pont Connect:Direct .

Vous pouvez transférer des données de fichier vers des données de message IBM MQ . Les messages IBM MQ peuvent être lus et utilisés par les applications. Les types de transfert de fichier à message suivants sont pris en charge:

- D'un fichier unique à un message unique. Aucun ID groupe IBM MQ n'est défini pour le message.
- D'un fichier unique à plusieurs messages, en fractionnant le fichier en messages d'une longueur donnée. Les messages ont tous le même ID groupe IBM MQ .
- D'un fichier unique à plusieurs messages, en fractionnant un fichier texte au niveau d'un délimiteur d'expression régulière Java . Les messages ont tous le même ID groupe IBM MQ .
- D'un fichier unique à plusieurs messages, en fractionnant un fichier binaire à l'aide d'un délimiteur hexadécimal. Les messages ont tous le même ID groupe IBM MQ .

Si vous souhaitez fractionner un fichier binaire en utilisant une séquence d'octets comme délimiteur, utilisez le paramètre **-sqdb** de la commande **fteCreateTransfer**. Pour plus d'informations, voir [Paramètre -sqdb](#).

Par défaut, les messages créés par un transfert de fichier à message sont persistants. Les messages peuvent être définis pour être non persistants ou pour avoir la valeur de persistance définie par la file d'attente de destination.

Si vous indiquez qu'un fichier est divisé en plusieurs messages, tous les messages créés à partir du fichier ont le même ID de groupe IBM MQ. Si vous ne spécifiez pas qu'un fichier est divisé en plusieurs messages, un seul message est créé à partir du fichier et l'ID de groupe IBM MQ n'est pas défini pour ce message.

Si vous transférez des fichiers vers des messages volumineux ou de nombreux messages de petite taille, vous devrez peut-être modifier certaines propriétés IBM MQ ou Managed File Transfer. Pour plus d'informations, voir [Conseils pour la définition des attributs MQ et des propriétés MFT associées à la taille des messages](#).

**Remarque :** Si la file d'attente de destination est une file d'attente en cluster ou un alias d'une file d'attente en cluster, vous obtiendrez un message d'erreur lors du transfert d'un fichier dans une file d'attente si la propriété d'agent `enableClusterQueueInputOutput` n'a pas été définie sur `true`. Pour plus d'informations, voir [Que faire si la file d'attente de destination est une file d'attente en cluster ou un alias d'une file d'attente en cluster](#)

### Tâches associées

[«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ, vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

[«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278](#)

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer**. Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

[«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280](#)

Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer**. Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

[«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer**.

[«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer**.

[«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286](#)

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de

sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287](#)

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Démarrage d'un nouveau transfert de fichiers», à la page 228](#)

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

[«Echec d'un transfert de fichier à message», à la page 290](#)

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

[Propriétés de message MQ définies par MFT sur les messages écrits dans les files d'attente de destination](#)  
[Conseils pour la définition des attributs MQ et des propriétés MFT associées à la taille des messages](#)

## Configuration d'un agent pour effectuer des transferts de fichier à message

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

### Pourquoi et quand exécuter cette tâche

Si vous tentez d'effectuer un transfert de fichier à message vers un agent de destination dont la propriété `enableQueueInputOutput` n'est pas définie sur `true`, le transfert échoue. Le message de journal de transfert publié dans le gestionnaire de file d'attente de coordination contient le message suivant:

```
BFGI00197E: An attempt to write to a queue was rejected by the destination agent. The agent must have enableQueueInputOutput=true set in the agent.properties file to support transferring to a queue.
```

Pour permettre à l'agent d'écrire et de lire dans les files d'attente, procédez comme suit:

### Procédure

1. Arrêtez l'agent de destination à l'aide de la commande **fteStopAgent** .
2. Editez le fichier `agent.properties` pour inclure la ligne `enableQueueInputOutput=true`.  
Le fichier `agent.properties` se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name`.
3. Facultatif : Editez le fichier `agent.properties` pour inclure la ligne `enableClusterQueueInputOutput=true`. Le fichier `agent.properties` se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/destination_agent_name`.
4. Démarrez l'agent cible à l'aide de la commande **fteStartAgent** .

### Concepts associés

[«Transfert de données de fichiers vers des messages», à la page 275](#)

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

## Tâches associées

«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer**. Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280

Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer**. Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer**.

«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer**.

«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

## Référence associée

### **fteStopAgent**

### **fteStartAgent**

Le fichier `MFT.agent.properties`

«Echec d'un transfert de fichier à message», à la page 290

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

## Exemple: Transfert d'un fichier unique vers un message unique

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer**. Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

## Pourquoi et quand exécuter cette tâche

Le fichier source est appelé `/tmp/single_record.txt` et se trouve sur le même système que l'agent source, AGENT\_NEPTUNE. L'agent source, AGENT\_NEPTUNE, utilise le gestionnaire de files d'attente QM\_NEPTUNE. L'agent de destination est AGENT\_VENUS et cet agent se connecte au gestionnaire de files d'attente QM\_VENUS. La file d'attente de destination, RECEIVING\_QUEUE, se trouve sur le gestionnaire de files d'attente QM\_MERCURY. QM\_MERCURY se trouve dans le même réseau IBM MQ que le gestionnaire de files d'attente QM\_VENUS, auquel il peut accéder.

## Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -dm QM_VENUS  
-dq RECEIVING_QUEUE@QM_MERCURY /tmp/single_record.txt
```

Si la file d'attente de destination se trouve sur un gestionnaire de files d'attente différent de celui utilisé par l'agent de destination, vous devez spécifier la valeur du paramètre **-dq** au format suivant: *nom\_file d'attente@nom\_gestionnaire\_files d'attente*. Si vous ne spécifiez pas *@nom\_gestionnaire\_files\_attente* dans la valeur, l'agent de destination suppose que la file d'attente de destination se trouve sur le gestionnaire de files d'attente de l'agent de destination. La seule exception est lorsque la propriété de l'agent de sortie `enableClusterQueueInputa` a été définie sur `true`. Dans ce cas, l'agent de destination utilise des procédures de résolution IBM MQ standard pour déterminer où se trouve la file d'attente.

L'agent source, AGENT\_NEPTUNE, lit les données du fichier `/tmp/single_record.txt` et les transfère à l'agent de destination, AGENT\_VENUS. L'agent de destination, AGENT\_VENUS, envoie les données à un message persistant dans la file d'attente `RECEIVING_QUEUE@QM_MERCURY`. Aucun ID groupe IBM MQ n'est défini pour le message.

## Concepts associés

[«Transfert de données de fichiers vers des messages», à la page 275](#)

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

## Tâches associées

[«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

[«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280](#)

Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer** . Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

[«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer** .

[«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer** .

[«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286](#)

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287](#)

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Démarrage d'un nouveau transfert de fichiers», à la page 228](#)

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

[«Echec d'un transfert de fichier à message», à la page 290](#)

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

## Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur

Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer** . Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

### Pourquoi et quand exécuter cette tâche

Le fichier source est appelé `/tmp/source.file` et sa taille est de 36 Ko. Le fichier source se trouve sur le même système que l'agent source AGENT\_NEPTUNE. L'agent source, AGENT\_NEPTUNE, se connecte au gestionnaire de files d'attente QM\_NEPTUNE. L'agent de destination est AGENT\_MERCURY, qui se connecte au gestionnaire de files d'attente QM\_MERCURY. La file d'attente de destination, RECEIVING\_QUEUE, se trouve également sur le gestionnaire de files d'attente QM\_MERCURY. Le transfert fractionne le fichier source en sections d'une taille de 1 Ko et écrit chacune de ces sections dans un message sur RECEIVING\_QUEUE.

### Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
                 -dq RECEIVING_QUEUE -qs 1K /tmp/source.file
```

L'agent source, AGENT\_NEPTUNE, lit les données du fichier `/tmp/source.file` et les transfère à l'agent cible, AGENT\_MERCURY. L'agent de destination, AGENT\_MERCURY, écrit les données dans trente-six messages persistants de 1 ko dans la file d'attente RECEIVING\_QUEUE@QM\_MERCURY. Ces messages ont tous le même ID de groupe IBM MQ et l'indicateur IBM MQ LAST\_MSG\_IN\_GROUP est défini pour le dernier message du groupe.

### Concepts associés

[«Transfert de données de fichiers vers des messages», à la page 275](#)

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .



## Tâches associées

«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ, vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer**. Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer**.

«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer**.

«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

«Démarrage d'un nouveau transfert de fichiers», à la page 228

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

## Référence associée

«Echec d'un transfert de fichier à message», à la page 290

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

## Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer**.

## Pourquoi et quand exécuter cette tâche

Le fichier est divisé en sections de longueur variable, dont chacune est écrite dans un message individuel. Le fichier texte est fractionné à chaque point où le texte du fichier correspond à une expression régulière donnée. Le fichier source est appelé `/tmp/names.text` et a le contenu suivant:

```
Jenny Jones,John Smith,Jane Brown
```

L'expression régulière qui indique où scinder le fichier est la virgule (,).

Le fichier source se trouve sur le même système que l'agent source `AGENT_NEPTUNE`, qui se connecte au gestionnaire de files d'attente `QM_NEPTUNE`. La file d'attente de destination, `RECEIVING_QUEUE`, se trouve sur le gestionnaire de files d'attente `QM_MERCURY`. `QM_MERCURY` est également le gestionnaire de files d'attente utilisé par l'agent de destination `AGENT_MERCURY`. Le transfert fractionne le fichier source en sections et écrit chacune de ces sections dans un message sur `RECEIVING_QUEUE`.

## Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY  
-dq RECEIVING_QUEUE -t text -dqdp postfix -dqdt "," /tmp/names.text
```

L'agent source, `AGENT_NEPTUNE`, lit les données du fichier `/tmp/names.text` et les transfère à l'agent cible, `AGENT_MERCURY`. L'agent de destination, `AGENT_MERCURY`, écrit les données dans trois messages persistants de la file d'attente `RECEIVING_QUEUE`. Ces messages ont tous le même ID de groupe IBM MQ et l'indicateur IBM MQ `LAST_MSG_IN_GROUP` est défini pour le dernier message du groupe.

Les données des messages sont les suivantes.

- Premier message:

```
Jenny Jones
```

- Deuxième message:

```
John Smith
```

- Troisième message:

```
Jane Brown
```

## Concepts associés

«Transfert de données de fichiers vers des messages», à la page 275

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

## Tâches associées

«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer** . Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files

d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280  
Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer**. Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer**.

«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

«Démarrage d'un nouveau transfert de fichiers», à la page 228

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

«Echec d'un transfert de fichier à message», à la page 290

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

[Expressions régulières utilisées par MFT](#)

## Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer**.

### Pourquoi et quand exécuter cette tâche

Transférez un fichier texte unique vers plusieurs messages d'une file d'attente. Le fichier est divisé en sections de longueur variable, dont chacune est écrite dans un message individuel. Le fichier texte est fractionné à chaque point où le texte du fichier correspond à une expression régulière donnée. Le fichier source est appelé `/tmp/customers.text` et a le contenu suivant:

```
Customer name: John Smith
Customer contact details: john@example.net
Customer number: 314
```

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

L'expression régulière qui indique où scinder le fichier est `Customer\snumber:\s\d+`, qui correspond au texte "Customer number: " suivi de n'importe quel nombre de chiffres. Les expressions régulières spécifiées sur la ligne de commande doivent être placées entre guillemets pour empêcher le shell de commandes d'évaluer l'expression régulière. L'expression régulière est évaluée en tant qu'expression régulière Java . Pour plus d'informations, voir [Expressions régulières utilisées par MFT](#).

Par défaut, le nombre de caractères pouvant correspondre à une expression régulière est défini sur cinq. L'expression régulière utilisée dans cet exemple correspond à des chaînes de plus de cinq caractères. Pour activer les correspondances de plus de cinq caractères, éditez le fichier de propriétés de l'agent pour inclure la propriété **maxDelimiterMatchLength**.

Par défaut, le texte qui correspond à l'expression régulière n'est pas inclus dans les messages. Pour inclure le texte qui correspond à l'expression régulière dans les messages, comme dans cet exemple, utilisez le paramètre **-qi** . Le fichier source se trouve sur le même système que l'agent source AGENT\_NEPTUNE, qui se connecte au gestionnaire de files d'attente QM\_NEPTUNE. La file d'attente de destination, RECEIVING\_QUEUE, se trouve sur le gestionnaire de files d'attente QM\_MERCURY. QM\_MERCURY est également le gestionnaire de files d'attente utilisé par l'agent de destination AGENT\_MERCURY. Le transfert fractionne le fichier source en sections et écrit chacune de ces sections dans un message sur RECEIVING\_QUEUE.

## Procédure

1. Arrêtez l'agent de destination à l'aide de la commande suivante:

```
fteStopAgent AGENT_MERCURY
```

2. Ajoutez la ligne suivante au fichier de propriétés de l'agent pour AGENT\_MERCURY:

```
maxDelimiterMatchLength=25
```

**Remarque :** L'augmentation de la valeur de **maxDelimiterMatchLength** peut réduire les performances.

3. Démarrez l'agent cible à l'aide de la commande suivante:

```
fteStartAgent AGENT_MERCURY
```

4. Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_MERCURY -dm QM_MERCURY
-dq RECEIVING_QUEUE
text -dqdt "Customer\snumber:\s\d+" -qi -dqdp postfix /tmp/customers.text
```

L'agent source, AGENT\_NEPTUNE, lit les données du fichier `/tmp/customers.text` et les transfère à l'agent cible, AGENT\_MERCURY. L'agent de destination, AGENT\_MERCURY, écrit les données dans trois messages persistants de la file d'attente RECEIVING\_QUEUE. Ces messages ont tous le même ID de groupe IBM MQ et l'indicateur IBM MQ LAST\_MSG\_IN\_GROUP est défini pour le dernier message du groupe.

Les données des messages sont les suivantes.

- Premier message:

```
Customer name: John Smith
```

```
Customer contact details: john@example.net
Customer number: 314
```

- Deuxième message:

```
Customer name: Jane Brown
Customer contact details: jane@example.com
Customer number: 42
```

- Troisième message:

```
Customer name: James Jones
Customer contact details: jjones@example.net
Customer number: 26
```

## Concepts associés

[«Transfert de données de fichiers vers des messages», à la page 275](#)

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

## Tâches associées

[«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

[«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278](#)

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer** . Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

[«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280](#)

Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer** . Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

[«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer** .

[«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286](#)

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287](#)

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Démarrage d'un nouveau transfert de fichiers», à la page 228](#)

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

[Le fichier MFT agent.properties](#)

[Expressions régulières utilisées par MFT](#)

## Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

### Pourquoi et quand exécuter cette tâche

Incluez le paramètre `-qmp true` dans la commande **fteCreateTransfer** . Dans cet exemple, l'ID utilisateur MQMD de l'utilisateur qui soumet la commande est `larmer`.

### Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM -qmp true
-t text /tmp/source_file.txt
```

Les propriétés de message IBM MQ du premier message écrit par l'agent de destination, AGENT\_SATURN, dans la file d'attente, MY\_QUEUE, sur le gestionnaire de files d'attente, MyQM, sont définies sur les valeurs suivantes:

```
usr.WMQFTETransferId=414cbaedefa234889d999a8ed09782395ea213ebbc9377cd
usr.WMQFTETransferMode=text
usr.WMQFTESourceAgent=AGENT_JUPITER
usr.WMQFTEDestinationAgent=AGENT_SATURN
usr.WMQFTEFileName=source_file.txt
usr.WMQFTEFileSize=1024
usr.WMQFTEFileLastModified=1273740879040
usr.WMQFTEFileIndex=0
usr.WMQFTEMqmdUser=larmer
```

### Concepts associés

«Transfert de données de fichiers vers des messages», à la page 275

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

### Tâches associées

«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer** . Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files

d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280  
Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer**. Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer**.

«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer**.

«Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

«Démarrage d'un nouveau transfert de fichiers», à la page 228

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

«Echec d'un transfert de fichier à message», à la page 290

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

Propriétés de message MQ définies par MFT sur les messages écrits dans les files d'attente de destination

## Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message

Les métadonnées définies par l'utilisateur sont définies en tant que propriété de message IBM MQ sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2.

### Pourquoi et quand exécuter cette tâche

Incluez les paramètres **-qmp true** et **-md account=123456** dans la commande **fteCreateTransfer** pour définir la propriété **usr.account** sur 123456 dans l'en-tête RFH2.

### Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq MY_QUEUE@MyQM  
-qmp true -md account=123456 /tmp/source_file.txt
```

Outre l'ensemble standard de propriétés de message IBM MQ , la propriété définie par l'utilisateur est définie dans l'en-tête du premier message écrit par l'agent de destination, AGENT\_SATURN, dans la file d'attente, MY\_QUEUE, sur le gestionnaire de files d'attente, MyQM. L'en-tête est défini sur la valeur suivante:

```
usr.account=123456
```

Le préfixe `usr` est ajouté au début du nom des métadonnées définies par l'utilisateur.

### Concepts associés

[«Transfert de données de fichiers vers des messages», à la page 275](#)

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

### Tâches associées

[«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

[«Exemple: Transfert d'un fichier unique vers un message unique», à la page 278](#)

Vous pouvez spécifier une file d'attente comme destination d'un transfert de fichier à l'aide du paramètre **-dq** avec la commande **fteCreateTransfer** . Le fichier source doit être inférieur à la longueur de message maximale définie dans la file d'attente de destination. La file d'attente de destination ne doit pas nécessairement se trouver sur le même gestionnaire de files d'attente que le gestionnaire de files d'attente auquel l'agent de destination se connecte, mais ces deux gestionnaires de files d'attente doivent pouvoir communiquer.

[«Exemple: Fractionnement d'un fichier unique en plusieurs messages par longueur», à la page 280](#)

Vous pouvez fractionner un fichier en plusieurs messages IBM MQ à l'aide du paramètre **-qs** de la commande **fteCreateTransfer** . Le fichier est divisé en sections de longueur fixe, dont chacune est écrite dans un message individuel.

[«Exemple: Fractionnement d'un fichier texte avec un délimiteur d'expression régulière et inclusion du délimiteur dans les messages», à la page 283](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée et incluez la correspondance d'expression régulière dans les messages résultants. Pour ce faire, utilisez les paramètres **-dqdt** et **-qi** de la commande **fteCreateTransfer** .

[«Exemple: division d'un fichier texte en plusieurs messages à l'aide d'un délimiteur d'expression régulière», à la page 281](#)

Transférez un fichier texte unique dans plusieurs messages en fractionnant le fichier à chaque correspondance d'une expression régulière Java donnée. Pour ce faire, utilisez le paramètre **-dqdt** de la commande **fteCreateTransfer** .

[«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286](#)

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

[«Démarrage d'un nouveau transfert de fichiers», à la page 228](#)

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

[Propriétés de message MQ définies par MFT sur les messages écrits dans les files d'attente de destination](#)



## Exemple: ajout d'une propriété de message définie par l'utilisateur pour un transfert de fichier à message

Si vous utilisez Managed File Transfer pour les transferts gérés de message à fichier, vous pouvez inclure une propriété de message définie par l'utilisateur pour le message résultant.

### Pourquoi et quand exécuter cette tâche

Vous pouvez utiliser l'une des méthodes suivantes pour définir une propriété de message personnalisée:

- Indiquez le paramètre **-md** dans la demande de transfert. Pour plus d'informations, voir la section «Exemple: Définition de propriétés définies par l'utilisateur sur un transfert de fichier à message», à la page 287.
- Utilisez une tâche Ant ; vous pouvez utiliser fte:filecopy ou fte:filemove. L'exemple suivant est une tâche fte:filecopy:

```
<project xmlns:fte="antlib:com.ibm.wmqfte.ant.taskdefs" default="complete">
<!-- Initialise the properties used in this script.-->

<target name="init" description="initialise task properties">
  <property name="src.file" value="/home/user/file1.bin"/>
  <property name="dst.queue" value="TEST.QUEUE@qm2"/>
  <fte:uuid property="job.name" length="8"
prefix="copyjob#"/>
</target>
<target name="step1" depends="init" description="transfer file">

<fte:filecopy cmdqm="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result">

<fte:metadata>
<fte:entry name="fileName" value="${FileName}"/>
</fte:metadata>

<fte:filespec srcfilespec="${src.file}" dstqueue="${dst.queue}"
dstmsgprops="true"/>

</fte:filecopy>

</target>
</project>
```

- Utilisez un moniteur de ressources et une substitution de variable. L'exemple suivant illustre un code XML de tâche de transfert:

```
<?xml version="1.0" encoding="UTF-8"?>
<monitor:monitor
xmlns:monitor="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance" version="5.00"
xsi:schemaLocation="https://www.ibm.com/xmlns/wmqfte/7.0.1/MonitorDefinition ./Monitor.xsd">
  <name>METADATA</name>
  <pollInterval units="minutes">5</pollInterval>
  <batch maxSize="5"/>
  <agent>AGENT1</agent>
  <resources>
    <directory recursionLevel="0">e:\temp</directory>
  </resources>
  <triggerMatch>
    <conditions>
      <allof>
        <condition>
          <fileMatch>
            <pattern>*.txt</pattern>
          </fileMatch>
        </condition>
      </allof>
    </conditions>
  </triggerMatch>
  <tasks>
    <task>
```

```

<name/>
<transfer>
  <request version="5.00"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
    <managedTransfer>
      <originator>
        <hostName>mqjason.raleigh.ibm.com.</hostName>
        <userID>administrator</userID>
      </originator>
      <sourceAgent QMgr="AGENTQM" agent="AGENT1"/>
      <destinationAgent QMgr="AGENTQM" agent="AGENT2"/>
      <transferSet priority="0">
        <metaDataSet>
          <metaData key="FileName">${FileName}</metaData>
        </metaDataSet>
        <item checksumMethod="MD5" mode="text">
          <source disposition="delete" recursive="false">
            <file>${FilePath}</file>
          </source>
          <destination type="queue">
            <queue persistent="true"
setMqProps="true">TEST.QUEUE@AGENTQM</queue>
          </destination>
        </item>
      </transferSet>
    </managedTransfer>
  </request>
</transfer>
</task>
</tasks>
<originator>
  <hostName>mqjason.raleigh.ibm.com.</hostName>
  <userID>administrator</userID>
</originator>
</monitor:monitor>

```

### Tâches associées

«Exemple: Définition des propriétés de message IBM MQ sur un transfert de fichier à message», à la page 286

Vous pouvez utiliser le paramètre **-qmp** dans la commande **fteCreateTransfer** pour indiquer si les propriétés de message IBM MQ sont définies sur le premier message écrit dans la file d'attente de destination par le transfert. Les propriétés de message IBM MQ permettent à une application de sélectionner des messages à traiter ou d'extraire des informations sur un message sans accéder aux en-têtes IBM MQ Message Descriptor (MQMD) ou MQRFH2 .

### Référence associée

tâche fte: filecopy Ant

fte: tâche Ant filemove

## Echec d'un transfert de fichier à message

Si un transfert de fichier à message échoue après que l'agent a commencé à écrire des données de fichier dans la file d'attente de destination, l'agent écrit un message dans la file d'attente pour indiquer à une application consommant les messages qu'un incident s'est produit.

Message écrit dans la file d'attente de destination en cas d'échec:

- Est vide
- A le même ID groupe IBM MQ que le message précédent écrit dans la file d'attente de destination par l'agent
- A l'indicateur IBM MQ LAST\_MSG\_IN\_GROUP défini
- Contient des propriétés de message IBM MQ supplémentaires, si les propriétés de message sont activées. Pour plus d'informations, voir la rubrique Propriétés des messagesMQ définies par MFT sur les messages écrits dans les files d'attente de destination.

## Exemple

Un transfert est demandé en exécutant la commande suivante:

```
fteCreateTransfer -sa AGENT_JUPITER -da AGENT_SATURN -dq RECEIVING_QUEUE  
-qmp true -qs 1K /tmp/source1.txt
```

Le fichier `source1.txt` est de 48 Ko. Le transfert fractionne ce fichier en messages de 1 Ko et écrit ces messages dans la file d'attente de destination `RECEIVING_QUEUE`.

Pendant que le transfert est en cours, une fois que l'agent a écrit 16 messages dans `RECEIVING_QUEUE`, un échec se produit au niveau de l'agent source.

L'agent écrit un message vide dans `RECEIVING_QUEUE`. En plus de l'ensemble standard de propriétés de message, le message vide possède les propriétés de message suivantes:

```
usr.WMQFTEResultCode = 40  
usr.WMQFTESupplement = BFGTR0036I: The transfer failed to complete successfully.
```

Depuis la IBM MQ 9.3.0, lorsqu'un transfert à partir d'un fichier échoue en raison d'une erreur de vérification de la taille du délimiteur, un seul message vide est envoyé. En outre, les propriétés utilisateur sont ajoutées à ce message si l'échec du transfert est dû au fait que le délimiteur a dépassé la taille définie sur l'agent de destination.

### Concepts associés

«Transfert de données de fichiers vers des messages», à la page 275

Vous pouvez utiliser la fonction de fichier à message de Managed File Transfer pour transférer des données d'un fichier vers un ou plusieurs messages d'une file d'attente IBM MQ .

### Tâches associées

«Configuration d'un agent pour effectuer des transferts de fichier à message», à la page 277

Par défaut, les agents ne peuvent pas effectuer de transferts de fichier à message ou de message à fichier. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`. Pour activer l'écriture dans les files d'attente en cluster IBM MQ , vous devez également définir la propriété d'agent `enableClusterQueueInputOutput` sur `true`.

«Démarrage d'un nouveau transfert de fichiers», à la page 228

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

### Référence associée

[Le fichier `MFT.agent.properties`](#)

[Propriétés de message MQ définies par MFT sur les messages écrits dans les files d'attente de destination](#)

## Transfert de données de messages vers des fichiers

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ , vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer .

Pour plus d'informations sur les transferts de fichier à message, voir «[Transfert de données de fichiers vers des messages](#)», à la page 275.



**Avertissement :** L'agent source d'un transfert de message à fichier ne peut pas être un agent de pont de protocole ou un agent de pont `Connect:Direct` .

Vous pouvez transférer des données de message IBM MQ dans un fichier. Les types de transfert de message vers fichier suivants sont pris en charge:

- D'un message unique vers un fichier unique

- De plusieurs messages vers un seul fichier
- De plusieurs messages avec le même ID groupe IBM MQ vers un seul fichier.
- De plusieurs messages à un seul fichier, y compris un délimiteur texte ou binaire entre les données de chaque message écrit dans le fichier.

Si vous transférez des fichiers à partir de messages volumineux ou de nombreux messages de petite taille, vous devrez peut-être modifier certaines propriétés IBM MQ ou Managed File Transfer . Pour plus d'informations, voir [Conseils pour la définition des attributs MQ et des propriétés MFT associées à la taille des messages](#).

Dans un transfert de message à fichier, l'agent source parcourt les messages à partir de la file d'attente source, contrairement à l'opération GET destructive dans les versions antérieures de IBM MQ. Les messages sont supprimés de la file d'attente source une fois que tous les messages (d'un groupe si le regroupement de messages est utilisé) ont été parcourus et que les données ont été écrites dans le fichier de destination. Ainsi, les messages peuvent rester dans la file d'attente source si un transfert échoue ou est annulé. En raison de cette modification, le droit de parcours (BROWSE) doit également être fourni avec le droit d'obtention (GET) pour l'exécution de transferts de messages dans des fichiers.

Managed File Transfer compare l'identificateur de transfert et la valeur de l'attribut groupId dans la charge XML de la demande de transfert. Si ces deux identificateurs sont équivalents, l'agent source utilise l'identificateur comme option de correspondance des identificateurs de message (et non comme option de correspondance des identificateurs de groupe) pour la première tentative MQGET effectuée sur la file d'attente d'entrée du transfert de message à fichier.

#### Tâches associées

«Exemple: configuration d'une ressource MFT», à la page 251

Vous pouvez spécifier une file d'attente IBM MQ comme ressource à surveiller par un moniteur de ressources à l'aide du paramètre **-mq** avec la commande **fteCreateMonitor** .

#### Référence associée

[Propriétés de message MQ lues par MFT à partir des messages des files d'attente source](#)

[Conseils pour la définition des attributs MQ et des propriétés MFT associées à la taille des messages](#)

## Configuration d'un agent pour effectuer des transferts de messages vers des fichiers

Par défaut, les agents ne peuvent pas effectuer de transferts de message à fichier ou de fichier à message. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`.

### Pourquoi et quand exécuter cette tâche

Si vous tentez d'exécuter un message de transfert de fichier à partir d'un agent source dont la propriété `enableQueueInputOutput` n'est pas définie sur `true`, le transfert échoue. Le message de journal de transfert publié dans le gestionnaire de file d'attente de coordination contient le message suivant:

```
BFGI00197E: An attempt to read from a queue was rejected by the source agent.
The agent must have enableQueueInputOutput=true set in the agent.properties file
to support transferring from a queue.
```

Pour permettre à l'agent d'écrire et de lire dans les files d'attente, procédez comme suit:

### Procédure

1. Arrêtez l'agent source à l'aide de la commande **fteStopAgent** .
2. Editez le fichier `agent.properties` pour inclure la ligne `enableQueueInputOutput=true`.  
Le fichier `agent.properties` se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/source_agent_name`.
3. Démarrez l'agent source à l'aide de la commande **fteStartAgent** .

## Concepts associés

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

## Tâches associées

[«Exemple: Transfert à partir d'une file d'attente vers un fichier unique», à la page 293](#)

Vous pouvez spécifier une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide du paramètre **-sq** avec la commande **fteCreateTransfer**.

[«Exemple: Transfert d'un groupe de messages d'une file d'attente vers un fichier unique», à la page 294](#)

Vous pouvez spécifier un groupe complet unique dans une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide des paramètres **-sq** et **-sqgi** avec la commande **fteCreateTransfer**.

[«Exemple: Insertion d'un délimiteur de texte avant les données de chaque message», à la page 296](#)

Lorsque vous effectuez un transfert en mode texte d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur de texte est inséré avant les données des messages individuels à l'aide des paramètres **-sq**, **-sqdt** et **-sqdp** avec la commande **fteCreateTransfer**.

[«Exemple: Insertion d'un délimiteur binaire après les données de chaque message», à la page 297](#)

Lors du transfert en mode binaire d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur binaire est inséré après les données des messages individuels à l'aide des paramètres **-sq**, **-sqdbet** et **-sqdp** avec la commande **fteCreateTransfer**.

[«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257](#)

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

[«Exemple: erreur d'un transfert de message à fichier à l'aide des propriétés de message IBM MQ», à la page 300](#)

Vous pouvez provoquer l'échec du transfert d'un message vers un fichier en définissant la propriété de message `usr.UserReturnCode` IBM MQ sur une valeur différente de zéro. Vous pouvez également spécifier des informations supplémentaires sur la raison de l'échec en définissant la propriété de message `usr.UserSupplement` IBM MQ.

## Référence associée

[Le fichier MFT agent.properties](#)

## Exemple: Transfert à partir d'une file d'attente vers un fichier unique

Vous pouvez spécifier une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide du paramètre **-sq** avec la commande **fteCreateTransfer**.

## Pourquoi et quand exécuter cette tâche

Les données source sont contenues dans trois messages de la file d'attente `START_QUEUE`. Cette file d'attente doit se trouver sur le gestionnaire de files d'attente de l'agent source, `QM_NEPTUNE`.

## Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE
                 -da AGENT_VENUS -df /out/three_to_one.txt
                 -sq START_QUEUE
```

Les données des messages de la file d'attente *START\_QUEUE* sont écrites dans le fichier */out/three\_to\_one.txt* sur le système sur lequel *AGENT\_VENUS* s'exécute.

### Concepts associés

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

### Tâches associées

[«Configuration d'un agent pour effectuer des transferts de messages vers des fichiers», à la page 292](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de message à fichier ou de fichier à message. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`.

[«Exemple: Transfert d'un groupe de messages d'une file d'attente vers un fichier unique», à la page 294](#)

Vous pouvez spécifier un groupe complet unique dans une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide des paramètres `-sq` et `-sqgi` avec la commande `fteCreateTransfer`.

[«Exemple: Insertion d'un délimiteur de texte avant les données de chaque message», à la page 296](#)

Lorsque vous effectuez un transfert en mode texte d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur de texte est inséré avant les données des messages individuels à l'aide des paramètres `-sq`, `-sqdt` et `-sqdp` avec la commande `fteCreateTransfer`.

[«Exemple: Insertion d'un délimiteur binaire après les données de chaque message», à la page 297](#)

Lors du transfert en mode binaire d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur binaire est inséré après les données des messages individuels à l'aide des paramètres `-sq`, `-sqdbet` et `-sqdp` avec la commande `fteCreateTransfer`.

[«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257](#)

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande `fteCreateMonitor`. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

[«Exemple: erreur d'un transfert de message à fichier à l'aide des propriétés de message IBM MQ», à la page 300](#)

Vous pouvez provoquer l'échec du transfert d'un message vers un fichier en définissant la propriété de message `usr.UserReturnCode` IBM MQ sur une valeur différente de zéro. Vous pouvez également spécifier des informations supplémentaires sur la raison de l'échec en définissant la propriété de message `usr.UserSupplement` IBM MQ.

### Référence associée

[Propriétés de message MQ lues par MFT à partir des messages des files d'attente source](#)

[\*\*fteCreateTransfer\*\*: démarrer un nouveau transfert de fichier](#)

## Exemple: Transfert d'un groupe de messages d'une file d'attente vers un fichier unique

Vous pouvez spécifier un groupe complet unique dans une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide des paramètres `-sq` et `-sqgi` avec la commande `fteCreateTransfer`.

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, il y a dix messages dans la file d'attente *START\_QUEUE*. Cette file d'attente doit se trouver sur le gestionnaire de files d'attente de l'agent source, *QM\_NEPTUNE*. Les trois premiers messages appartiennent à un groupe avec l' IBM MQ ID groupe `41424b3ef3a22020202020202020202020202020202020201111`; ce groupe n'est pas un groupe complet. Les cinq messages suivants appartiennent à un groupe avec l' IBM MQ

ID groupe 41424b3ef3a22020202020202020202020202020202020202222; ce groupe est complet. Les deux messages restants appartiennent à un groupe avec l' IBM MQ ID groupe 41424b3ef3a220202020202020202020202020202020203333; ce groupe est complet.

## Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS  
-df /out/group.txt -sqgi -sq START_QUEUE
```

Les données des messages appartenant au premier groupe complet de la file d'attente *START\_QUEUE*, le groupe avec l' IBM MQ ID groupe 41424b3ef3a220202020202020202020202020202020202222, sont écrites dans le fichier */out/group.txt* sur le système où *AGENT\_VENUS* s'exécute.

## Concepts associés

«Transfert de données de messages vers des fichiers», à la page 291

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

## Tâches associées

«Configuration d'un agent pour effectuer des transferts de messages vers des fichiers», à la page 292

Par défaut, les agents ne peuvent pas effectuer de transferts de message à fichier ou de fichier à message. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`.

«Exemple: Transfert à partir d'une file d'attente vers un fichier unique», à la page 293

Vous pouvez spécifier une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide du paramètre **-sq** avec la commande **fteCreateTransfer**.

«Exemple: Insertion d'un délimiteur de texte avant les données de chaque message», à la page 296

Lorsque vous effectuez un transfert en mode texte d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur de texte est inséré avant les données des messages individuels à l'aide des paramètres **-sq**, **-sqdt** et **-sqdp** avec la commande **fteCreateTransfer**.

«Exemple: Insertion d'un délimiteur binaire après les données de chaque message», à la page 297

Lors du transfert en mode binaire d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur binaire est inséré après les données des messages individuels à l'aide des paramètres **-sq**, **-sqdbet** et **-sqdp** avec la commande **fteCreateTransfer**.

«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

«Exemple: erreur d'un transfert de message à fichier à l'aide des propriétés de message IBM MQ», à la page 300

Vous pouvez provoquer l'échec du transfert d'un message vers un fichier en définissant la propriété de message `usr.UserReturnCode` IBM MQ sur une valeur différente de zéro. Vous pouvez également spécifier des informations supplémentaires sur la raison de l'échec en définissant la propriété de message `usr.UserSupplement` IBM MQ.

## Référence associée

**fteCreateTransfer**: démarrer un nouveau transfert de fichier

## Exemple: Insertion d'un délimiteur de texte avant les données de chaque message

Lorsque vous effectuez un transfert en mode texte d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur de texte est inséré avant les données des messages individuels à l'aide des paramètres **-sq**, **-sqdt** et **-sqdp** avec la commande **fteCreateTransfer**.

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, il y a quatre messages dans la file d'attente `START_QUEUE`. Cette file d'attente se trouve sur le gestionnaire de files d'attente de l'agent source, `QM_NEPTUNE`. Le délimiteur de texte à insérer avant les données de chaque message peut être exprimé sous la forme d'une chaîne littérale Java, par exemple: `\n\u002D\u002D\u002D\n`.

### Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/output.txt
-t text -sqdt "\n\u002D\u002D\u002D\n" -sqdp prefix -sq START_QUEUE
```

Le délimiteur de texte est ajouté au début des données de chacun des quatre messages de `START_QUEUE` par l'agent source, `AGENT_NEPTUNE`. Ces données sont écrites dans le fichier de destination, `/out/output.txt`.

### Concepts associés

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

### Tâches associées

[«Configuration d'un agent pour effectuer des transferts de messages vers des fichiers», à la page 292](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de message à fichier ou de fichier à message. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`.

[«Exemple: Transfert à partir d'une file d'attente vers un fichier unique», à la page 293](#)

Vous pouvez spécifier une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide du paramètre **-sq** avec la commande **fteCreateTransfer**.

[«Exemple: Transfert d'un groupe de messages d'une file d'attente vers un fichier unique», à la page 294](#)

Vous pouvez spécifier un groupe complet unique dans une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide des paramètres **-sq** et **-sqgi** avec la commande **fteCreateTransfer**.

[«Exemple: Insertion d'un délimiteur binaire après les données de chaque message», à la page 297](#)

Lors du transfert en mode binaire d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur binaire est inséré après les données des messages individuels à l'aide des paramètres **-sq**, **-sqdbet** et **-sqdp** avec la commande **fteCreateTransfer**.

[«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257](#)

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

[«Exemple: erreur d'un transfert de message à fichier à l'aide des propriétés de message IBM MQ», à la page 300](#)



Vous pouvez provoquer l'échec du transfert d'un message vers un fichier en définissant la propriété de message `usr.UserReturnCode` IBM MQ sur une valeur différente de zéro. Vous pouvez également spécifier des informations supplémentaires sur la raison de l'échec en définissant la propriété de message `usr.UserSupplement` IBM MQ .

### Référence associée

**`fteCreateTransfer`**: démarrer un nouveau transfert de fichier

## Exemple: Insertion d'un délimiteur binaire après les données de chaque message

Lors du transfert en mode binaire d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur binaire est inséré après les données des messages individuels à l'aide des paramètres **-sq**, **-sqdbet** **-sqdp** avec la commande **`fteCreateTransfer`** .

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, il y a trois messages dans la file d'attente `START_QUEUE`. Cette file d'attente se trouve sur le gestionnaire de files d'attente de l'agent source, `QM_NEPTUNE`. Le délimiteur binaire à insérer après les données de chaque message doit être exprimé sous la forme d'une liste d'octets hexadécimaux séparés par des virgules, par exemple: `x34,xE7,xAE`.

### Procédure

Entrez la commande suivante :

```
fteCreateTransfer -sa AGENT_NEPTUNE -sm QM_NEPTUNE -da AGENT_VENUS -df /out/binary.file  
-sqdp postfix -sqdb x34,xE7,xAE -sq START_QUEUE
```

Le délimiteur binaire est ajouté aux données de chacun des trois messages de `START_QUEUE` par l'agent source, `AGENT_NEPTUNE`. Ces données sont écrites dans le fichier de destination, `/out/binary.file`.

### Concepts associés

«Transfert de données de messages vers des fichiers», à la page 291

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ , vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer .

### Tâches associées

«Configuration d'un agent pour effectuer des transferts de messages vers des fichiers», à la page 292

Par défaut, les agents ne peuvent pas effectuer de transferts de message à fichier ou de fichier à message. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`.

«Exemple: Transfert à partir d'une file d'attente vers un fichier unique», à la page 293

Vous pouvez spécifier une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide du paramètre **-sq** avec la commande **`fteCreateTransfer`** .

«Exemple: Transfert d'un groupe de messages d'une file d'attente vers un fichier unique», à la page 294

Vous pouvez spécifier un groupe complet unique dans une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide des paramètres **-sq** et **-sqgi** avec la commande **`fteCreateTransfer`** .

«Exemple: Insertion d'un délimiteur de texte avant les données de chaque message», à la page 296

Lorsque vous effectuez un transfert en mode texte d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur de texte est inséré avant les données des messages individuels à l'aide des paramètres **-sq**, **-sqdt** et **-sqdp** avec la commande **`fteCreateTransfer`** .

«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

«Exemple: erreur d'un transfert de message à fichier à l'aide des propriétés de message IBM MQ», à la page 300

Vous pouvez provoquer l'échec du transfert d'un message vers un fichier en définissant la propriété de message `usr.UserReturnCode` IBM MQ sur une valeur différente de zéro. Vous pouvez également spécifier des informations supplémentaires sur la raison de l'échec en définissant la propriété de message `usr.UserSupplement` IBM MQ.

### Référence associée

**fteCreateTransfer**: démarrer un nouveau transfert de fichier

## Surveillance d'une file d'attente et utilisation de la substitution de variable

Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor**. La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent source est appelé `AGENT_VENUS`, qui se connecte à `QM_VENUS`. La file d'attente surveillée par `AGENT_VENUS` est appelée `START_QUEUE` et se trouve sur `QM_VENUS`. L'agent interroge la file d'attente toutes les 30 minutes.

Lorsqu'un groupe complet de messages est écrit dans la file d'attente, la tâche de surveillance envoie le groupe de messages dans un fichier à l'un des agents de destination, qui se connectent tous au gestionnaire de files d'attente `QM_MARS`. Le nom du fichier vers lequel le groupe de messages est transféré est défini par la IBM MQ propriété de message `usr.fileName` sur le premier message du groupe. Le nom de l'agent auquel le groupe de messages est envoyé est défini par la IBM MQ propriété de message `usr.toAgent` sur le premier message du groupe. Si l'en-tête `usr.toAgent` n'est pas défini, la valeur par défaut à utiliser pour l'agent de destination est `AGENT_MAGENTA`.

Lorsque vous spécifiez `useGroups="true"`, si vous ne spécifiez pas également `groupId="{GROUPID}"`, le transfert prend simplement le premier message de la file d'attente. Par exemple, si vous utilisez la substitution de variable pour générer le `fileName`, il est possible que le contenu de `a.txt` ne soit pas correct. En effet, `fileName` est généré par le moniteur, mais le transfert reçoit en fait un message qui n'est pas celui qui doit générer le fichier appelé `fileName`.

## Procédure

1. Créez le code XML de tâche qui définit la tâche que le moniteur exécute lorsqu'il est déclenché.

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="{toAgent}" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="{GROUPID}">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/{fileName}.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

```
</transferSet>
</managedTransfer>
</request>
```

Les variables qui sont remplacées par les valeurs des en-têtes de message IBM MQ sont mises en évidence en **gras**. Cette tâche XML est sauvegardée dans le fichier `/home/USER1/task.xml`

2. Créez un moniteur de ressources pour surveiller la file d'attente `START_QUEUE`.

Soumettez la commande suivante :

```
fteCreateMonitor -ma AGENT_VENUS -mm QM_VENUS -mq START_QUEUE
                 -mn myMonitor -mt /home/USER1/task.xml
                 -tr completeGroups -pi 30 -pu minutes -dv toAgent=AGENT_MAGENTA
```

3. Un utilisateur ou un programme écrit un groupe de messages dans la file d'attente `START_QUEUE`. Les propriétés de message IBM MQ suivantes sont définies pour le premier message de ce groupe:

```
usr.fileName=larmer
usr.toAgent=AGENT_VIOLET
```

4. Le moniteur est déclenché lorsque le groupe complet est écrit. L'agent remplace les propriétés de message IBM MQ dans le code XML de la tâche.

Cela entraîne la transformation du code XML de la tâche en:

```
<?xml version="1.0" encoding="UTF-8" ?>
<request version="4.00"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="FileTransfer.xsd">
  <managedTransfer>
    <originator>
      <hostName>reportserver.com</hostName>
      <userID>USER1</userID>
    </originator>
    <sourceAgent agent="AGENT_VENUS" QMgr="QM_VENUS" />
    <destinationAgent agent="AGENT_VIOLET" QMgr="QM_MARS" />
    <transferSet>
      <item mode="binary" checksumMethod="none">
        <source>
          <queue useGroups="true" groupId="<b>${GROUPID}</b>">START_QUEUE</queue>
        </source>
        <destination type="file" exist="overwrite">
          <file>/reports/larmer.rpt</file>
        </destination>
      </item>
    </transferSet>
  </managedTransfer>
</request>
```

## Résultats

Le transfert défini par la tâche XML est effectué. Le groupe complet de messages lus à partir de la file d'attente `START_QUEUE` par `AGENT_VENUS` est écrit dans un fichier appelé `/reports/larmer.rpt` sur le système où s'exécute `AGENT_VIOLET`.

## Que faire ensuite

### Transfert de chaque message dans un fichier distinct

Si vous souhaitez surveiller une file d'attente et que chaque message soit transféré dans un fichier distinct, vous pouvez utiliser une technique similaire à celle décrite précédemment dans cette rubrique.

1. Créez le moniteur comme décrit précédemment, en spécifiant le paramètre **-tr completeGroups** dans la commande **fteCreateMonitor**.
2. Dans le code XML de la tâche, indiquez ce qui suit:

```
<queue useGroups="true" groupId="<b>${GROUPID}</b>">START_QUEUE</queue>
```

Toutefois, lorsque vous placez les messages dans la file d'attente source, ne les placez pas dans un groupe IBM MQ . Ajoutez des propriétés de message IBM MQ à chaque message. Par exemple, spécifiez la propriété `usr.filename` avec une valeur de nom de fichier unique pour chaque message. Ainsi, Managed File Transfer Agent traite chaque message de la file d'attente source comme un groupe distinct.

### Concepts associés

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ , vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer .

[«Personnalisation des tâches du moniteur de ressources MFT avec la substitution de variables», à la page 252](#)

Lorsque les conditions de déclenchement d'un moniteur de ressources actif sont satisfaites, la tâche définie est appelée. En plus d'appeler la tâche de transfert ou de commande avec le même agent de destination ou le même nom de fichier de destination à chaque fois, vous pouvez également modifier la définition de la tâche lors de l'exécution. Pour ce faire, vous devez insérer des noms de variable dans le fichier XML de définition de tâche. Lorsque le moniteur détermine que les conditions de déclenchement sont satisfaites et que la définition de tâche contient des noms de variable, il remplace les noms de variable par les valeurs de variable, puis appelle la tâche.

[Que faire si les fichiers de destination créés par un transfert démarré par un moniteur de ressources de file d'attente contiennent des données incorrectes](#)

### Tâches associées

[«Configuration des tâches de surveillance MFT pour lancer des commandes et des scripts», à la page 245](#)

Les moniteurs de ressources ne sont pas limités à l'exécution de transferts de fichiers en tant que tâche associée. Vous pouvez également configurer le moniteur pour qu'il appelle d'autres commandes à partir de l'agent de surveillance, notamment des programmes exécutables, des scripts Ant ou des travaux JCL. Pour appeler des commandes, éditez le fichier XML de définition de tâche de contrôle pour inclure un ou plusieurs éléments de commande avec les paramètres d'appel de commande correspondants, tels que les arguments et les propriétés.

[«Exemple: configuration d'une ressource MFT», à la page 251](#)

Vous pouvez spécifier une file d'attente IBM MQ comme ressource à surveiller par un moniteur de ressources à l'aide du paramètre `-mq` avec la commande `fteCreateMonitor` .

### Référence associée

[`fteCreateMonitor`: créer un moniteur de ressources MFT](#)

[Propriétés de message MQ lues par MFT à partir des messages des files d'attente source](#)

## Exemple: erreur d'un transfert de message à fichier à l'aide des propriétés de message IBM MQ

Vous pouvez provoquer l'échec du transfert d'un message vers un fichier en définissant la propriété de message `usr.UserReturnCode` IBM MQ sur une valeur différente de zéro. Vous pouvez également spécifier des informations supplémentaires sur la raison de l'échec en définissant la propriété de message `usr.UserSupplement` IBM MQ .

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, un transfert est en cours entre la file d'attente `INPUT_QUEUE` et le fichier `/home/user/output.file`.

Un utilisateur crée des messages et les place dans la file d'attente `INPUT_QUEUE`. L'agent source consomme des messages de la file d'attente `INPUT_QUEUE` et envoie les données de transfert à l'agent cible. L'agent de destination écrit ces données dans le fichier `/home/user/output.file`.

L'utilisateur qui écrit des messages dans la file d'attente `INPUT_QUEUE` souhaite arrêter le transfert en cours et supprimer les données qui ont déjà été écrites dans le fichier de destination.

## Procédure

1. L'utilisateur écrit un message dans la file d'attente INPUT\_QUEUE dont les propriétés de message IBM MQ suivantes sont définies:

```
usr.UserReturnCode=1
usr.UserSupplement="Cancelling transfer - sent wrong data."
```

2. L'agent source lit les propriétés de message IBM MQ et arrête de traiter les messages de la file d'attente. L'agent de destination supprime toutes les données de fichier qui ont été écrites dans le répertoire de destination.
3. L'agent source envoie un message de journal de transfert au gestionnaire de files d'attente de coordination signalant l'échec du transfert.

Le message contient les informations suivantes:

```
<?xml version="1.0" encoding="UTF-8"?>
<transaction version="1.00"
  ID="414d5120514d3120202020202020202020202020207e970d4920008702" agentRole="sourceAgent"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="TransferLog.xsd"
  xmlns="">
  <action time="2008-11-02T21:28:09.593Z">progress</action>
  <sourceAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </sourceAgent>
  <destinationAgent agent="FTEAGENT" QMgr="QM1">
    <systemInfo architecture="x86" name="Windows 7"
      version="6.1 build 7601 Service Pack 1" />
  </destinationAgent>
  <originator>
    <hostName>reportserver.com</hostName>
    <userID>USER1</userID>
    <mqmdUserID>USER1 </mqmdUserID>
  </originator>
  <transferSet index="0" size="1"
    startTime="2008-11-02T21:28:09.281Z"
    total="1">
    <item mode="binary">
      <source>
        <queue>INPUT_QUEUE@QM1</queue>
      </source>
      <destination exist="error">
        <file>/home/user/output.file</file>
      </destination>
      <status resultCode="1">
        <supplement>Cancelling transfer - sent wrong data.</supplement>
      </status>
    </item>
  </transferSet>
</transaction>
```

## Concepts associés

[«Transfert de données de messages vers des fichiers», à la page 291](#)

La fonction de message à fichier de Managed File Transfer vous permet de transférer des données d'un ou de plusieurs messages d'une file d'attente IBM MQ vers un fichier, un fichier (sous z/OS), ou un espace fichier utilisateur. Si vous disposez d'une application qui crée ou traite des messages IBM MQ, vous pouvez utiliser la fonction de message à fichier de Managed File Transfer pour transférer ces messages vers un fichier sur n'importe quel système de votre réseau Managed File Transfer.

## Tâches associées

[«Configuration d'un agent pour effectuer des transferts de messages vers des fichiers», à la page 292](#)

Par défaut, les agents ne peuvent pas effectuer de transferts de message à fichier ou de fichier à message. Pour activer cette fonction, vous devez définir la propriété d'agent `enableQueueInputOutput` sur `true`.

[«Exemple: Transfert à partir d'une file d'attente vers un fichier unique», à la page 293](#)

Vous pouvez spécifier une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide du paramètre **-sq** avec la commande **fteCreateTransfer** .

«Exemple: Transfert d'un groupe de messages d'une file d'attente vers un fichier unique», à la page 294  
Vous pouvez spécifier un groupe complet unique dans une file d'attente IBM MQ comme source d'un transfert de fichier à l'aide des paramètres **-sq** et **-sqgi** avec la commande **fteCreateTransfer** .

«Exemple: Insertion d'un délimiteur de texte avant les données de chaque message», à la page 296  
Lorsque vous effectuez un transfert en mode texte d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur de texte est inséré avant les données des messages individuels à l'aide des paramètres **-sq**, **-sqdt** et **-sqdp** avec la commande **fteCreateTransfer** .

«Exemple: Insertion d'un délimiteur binaire après les données de chaque message», à la page 297  
Lors du transfert en mode binaire d'une file d'attente source vers un fichier, vous pouvez indiquer qu'un délimiteur binaire est inséré après les données des messages individuels à l'aide des paramètres **-sq**, **-sqdbet** et **-sqdp** avec la commande **fteCreateTransfer** .

«Surveillance d'une file d'attente et utilisation de la substitution de variable», à la page 257  
Vous pouvez surveiller une file d'attente et transférer des messages de la file d'attente surveillée vers un fichier à l'aide de la commande **fteCreateMonitor** . La valeur de toute propriété de message IBM MQ dans le premier message à lire dans la file d'attente surveillée peut être remplacée dans la définition XML de tâche et utilisée pour définir le comportement de transfert.

### Référence associée

Propriétés de message MQ lues par MFT à partir des messages des files d'attente source

## Le pont de protocole

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT , dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

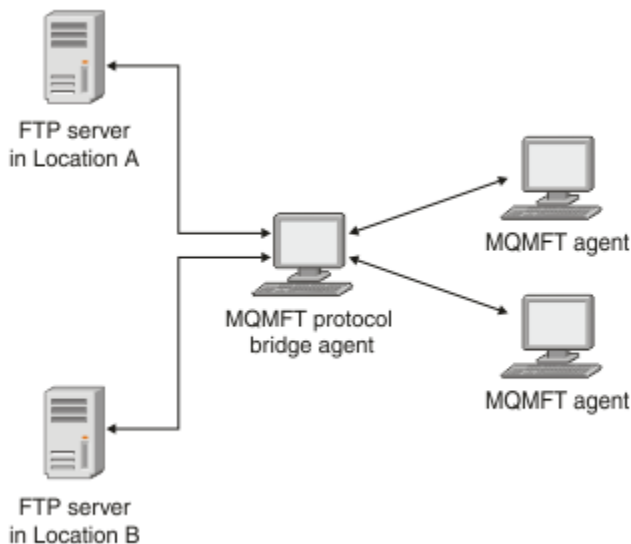
Le pont de protocole est disponible dans le cadre du composant Service de Managed File Transfer. Vous pouvez avoir plusieurs agents dédiés sur un seul système exécutant MFT qui se connectent à différents serveurs de fichiers.

Vous pouvez utiliser un agent de pont de protocole pour transférer des fichiers vers plusieurs noeuds finaux simultanément. MFT fournit un fichier appelé `ProtocolBridgeProperties.xml` que vous pouvez éditer pour définir les différents serveurs de fichiers de protocole vers lesquels vous souhaitez transférer des fichiers. La commande **fteCreateBridgeAgent** ajoute les détails du serveur de fichiers de protocole par défaut à `ProtocolBridgeProperties.xml` pour vous. Ce fichier est décrit dans Protocol bridge properties file format.

Vous pouvez utiliser l'agent de pont de protocole pour effectuer les actions suivantes:

- Téléchargez des fichiers depuis le réseau MFT vers un serveur distant à l'aide de FTP, FTPS ou SFTP.
- Téléchargement de fichiers depuis un serveur distant, à l'aide de FTP, FTPS ou SFTP, sur le réseau MFT

**Remarque :** L'agent de pont de protocole peut prendre en charge uniquement les serveurs FTP, FTPS ou SFTP qui permettent d'accéder aux fichiers par leur chemin d'accès absolu. Si un chemin de fichier relatif est spécifié dans une demande de transfert, l'agent de pont de protocole tente de convertir le chemin relatif en chemin de fichier absolu en fonction du répertoire de base utilisé pour la connexion au serveur de protocole. Les serveurs de protocole qui autorisent l'accès aux fichiers basés uniquement sur le répertoire en cours ne sont pas pris en charge par l'agent de pont de protocole.



Le diagramme montre deux serveurs FTP, à des emplacements différents. Les serveurs FTP sont utilisés pour échanger des fichiers avec les agents Managed File Transfer . L'agent de pont de protocole se trouve entre les serveurs FTP et le reste du réseau MFT et est configuré pour communiquer avec les deux serveurs FTP.

Vérifiez que vous disposez d'un autre agent dans votre réseau MFT en plus de l'agent de pont de protocole. L'agent de pont de protocole est un pont vers le serveur FTP, FTPS ou SFTP uniquement et n'écrit pas les fichiers transférés sur le disque local. Si vous souhaitez transférer des fichiers vers ou depuis le serveur FTP, FTPS ou SFTP, vous devez utiliser l'agent de pont de protocole comme destination ou source du transfert de fichiers (représentant le serveur FTP, FTPS ou SFTP) et un autre agent standard comme source ou destination correspondante.

Lorsque vous transférez des fichiers à l'aide du pont de protocole, le pont doit avoir le droit de lire le répertoire source ou de destination contenant les fichiers que vous souhaitez transférer. Par exemple, si vous souhaitez transférer des fichiers à partir du répertoire `/home/fte/bridge` qui dispose de droits d'exécution (`d -- x -- x -- x`) uniquement, les transferts que vous tentez d'effectuer à partir de ce répertoire échouent avec le message d'erreur suivant:

```
BFGBR0032E: Attempt to read filename from the protocol file server
has failed with server error 550. Failed to open file.
```

## Configuration d'un agent de pont de protocole

Un agent de pont de protocole est similaire à un agent MFT standard. Créez un agent de pont de protocole à l'aide de la commande **fteCreateBridgeAgent** . Vous pouvez configurer un agent de pont de protocole à l'aide du fichier `ProtocolBridgeProperties.xml` , qui est décrit dans [Protocol bridge properties file format](#) . Si vous utilisez une version antérieure, configurez l'agent à l'aide des propriétés de pont de protocole spécifiques décrites dans [Propriétés d'agent avancées: Pont de protocole](#) et [Propriétés d'agent avancées: Journalisation de l'agent de pont de protocole](#) . Pour toutes les versions, vous pouvez également configurer un mappage de données d'identification comme décrit dans [«Mappage des données d'identification pour un serveur de fichiers»](#), à la page 311. Une fois que vous avez configuré un agent de pont de protocole pour un serveur de fichiers de protocole particulier, vous ne pouvez utiliser cet agent qu'à cette fin.

## Reprise du pont de protocole

Si l'agent de pont de protocole ne parvient pas à se connecter au serveur de fichiers car ce dernier n'est pas disponible, toutes les demandes de transfert de fichiers sont mises en file d'attente jusqu'à ce que le serveur de fichiers soit disponible. Si l'agent de pont de protocole ne parvient pas à se connecter au serveur de fichiers car l'agent utilise des données d'identification incorrectes, le transfert échoue et le

message du journal de transfert reflète cette erreur. Si l'agent de pont de protocole est arrêté pour une raison quelconque, tous les transferts de fichiers demandés sont conservés et continuent lorsque le pont de protocole est redémarré.

Lors du transfert de fichiers, les fichiers sont généralement écrits en tant que fichiers temporaires sur la destination et sont ensuite renommés lorsque le transfert est terminé. Toutefois, si la destination de transfert est un serveur de fichiers de protocole configuré en écriture limitée (les utilisateurs peuvent télécharger des fichiers sur le serveur de fichiers de protocole mais ne peuvent pas modifier ces fichiers téléchargés de quelque manière que ce soit ; les utilisateurs peuvent écrire une seule fois), les fichiers transférés sont écrits directement sur la destination. Cela signifie que si un problème se produit lors du transfert, les fichiers partiellement écrits restent sur le serveur de fichiers de protocole de destination et Managed File Transfer ne peut pas supprimer ou éditer ces fichiers. Dans ce cas, le transfert échoue.

### Tâches associées

[«Exemple: Comment configurer un agent de pont de protocole pour utiliser des données d'identification de clé privée avec un serveur SFTP UNIX», à la page 316](#)

Cet exemple explique comment générer et configurer le fichier `ProtocolBridgeCredentials.xml`. Cet exemple est typique et les détails peuvent varier en fonction de votre plateforme, mais les principes restent les mêmes.

[«Définition des propriétés des serveurs de fichiers de protocole à l'aide du fichier ProtocolBridgeProperties.xml», à la page 304](#)

Définissez les propriétés d'un ou de plusieurs serveurs de fichiers de protocole vers et depuis lesquels vous souhaitez transférer des fichiers à l'aide du fichier `ProtocolBridgeProperties.xml`, fourni par Managed File Transfer dans le répertoire de configuration de l'agent.

### Référence associée

[fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT\)](#)

[«Mappage des données d'identification pour un serveur de fichiers», à la page 311](#)

Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole ou en écrivant votre propre exit utilisateur. Managed File Transfer fournit un exemple d'exit utilisateur qui effectue le mappage des données d'identification de l'utilisateur.

[Interface ProtocolBridgeCredentialExit.java](#)

[Exemple d'exit utilisateur de données d'identification de pont de protocole](#)

[Prise en charge du serveur FTPS par le pont de protocole](#)

## Définition des propriétés des serveurs de fichiers de protocole à l'aide du fichier `ProtocolBridgeProperties.xml`

Définissez les propriétés d'un ou de plusieurs serveurs de fichiers de protocole vers et depuis lesquels vous souhaitez transférer des fichiers à l'aide du fichier `ProtocolBridgeProperties.xml`, fourni par Managed File Transfer dans le répertoire de configuration de l'agent.

### Pourquoi et quand exécuter cette tâche

La commande **fteCreateBridgeAgent** crée le fichier `ProtocolBridgeProperties.xml` dans le répertoire de configuration de l'agent `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`. La commande crée également une entrée dans le fichier pour le serveur de fichiers de protocole par défaut, si une valeur par défaut a été spécifiée lors de l'exécution de la commande.

Le message BFGCL0392I indique l'emplacement du fichier `ProtocolBridgeProperties.xml`.

```
<?xml version="1.0" encoding="IBM-1047"?>
<!--
This ProtocolBridgeProperties.xml file determines the protocol servers that will be accessed by
the
MQMFT protocol bridge agent.
```



Each protocol server is defined using either a <tns:ftpServer>, <tns:ftpsServer>, or <tns:sftpServer> element - depending on the protocol used to communicate with the server. When the protocol bridge agent participates in a managed file transfer it will determine which server to use based on the prefix (if any) present on the file path. For example a file path of 'server1:/home/user/file.txt' would be interpreted as a request to transfer /home/user/file.txt using 'server1'. The server name is compared to the 'name' attribute of each <tns:ftpServer>, <tns:ftpsServer> or <tns:sftpServer> element in this XML document and the first match is used to determine which protocol server the protocol bridge agent will connect to. If no match is found then the managed file transfer operation will fail.

If a file path is not prefixed with a server name, for example '/home/user/file.txt' then this XML document can specify a default server to use for the managed file transfer. To specify a default server use the <tns:defaultServer> element as the first element inside the <tns:serverProperties> element. The default server will be used whenever the protocol bridge agent participates in a managed file transfer for file names which do not specify a prefix.

An optional <tns:limits> element can be specified within each server definition. This element contains attributes that govern the amount of resources used by each defined server.

An optional <tns:credentialsFile> element can be specified within each serverProperties definition. This element contains a path to a file containing credentials to be used when connecting to defined servers.

An example ProtocolBridgeProperties.xml file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">

  <tns:credentialsFile path="$HOME/ProtocolBridgeCredentials.xml" />

  <tns:defaultServer name="myFTPserver" />

  <tns:ftpServer name="myFTPserver" host="windows.hursley.ibm.com" port="1234"
platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  listFormat="unix" limitedWrite="false">

    <tns:limits maxListFileNames="100" maxListDirectoryLevels="999999999"
      maxReconnectRetry="2" reconnectWaitPeriod="10"
      maxSessions="60" socketTimeout="30" />

  </tns:ftpServer>

  <tns:ftpsServer name="myFTPSserver" host="unix.hursley.ibm.com" platform="unix"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false" ftpsType="explicit"
  trustStore="C:\FTE\keystores\myFTPSserver\FTPSKeyStore.jks"
  trustStorePassword="password">

    <tns:limits maxReconnectRetry="10" connectionTimeout="10"/>

  </tns:ftpsServer>

  <tns:sftpServer name="mySFTPserver" host="windows.hursley.ibm.com" platform="windows"
  timeZone="Europe/London" locale="en_GB" fileEncoding="UTF-8"
  limitedWrite="false">

    <tns:limits connectionTimeout="60"/>

  </tns:sftpServer>
</tns:serverProperties>
```

This example shows the outermost <tns:serverProperties> element which must exist for the document to be valid, an optional <tns:defaultServer> element, as well as definitions for an FTP, FTPS and SFTP server.

The attributes of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements determine the characteristics of the connection established to the server. These attributes correspond to

the command  
line parameters for the 'fteCreateBridgeAgent' command.

The following attributes are valid for all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: name, host, port, platform, fileEncoding, limitedWrite and controlEncoding.

The following attributes are valid for the <tns:ftpServer> and <tns:ftpsServer> elements: timeZone, locale, listFormat, listFileRecentDateFormat, listFileOldDateFormat, and monthShortNames.

The following attributes are valid for the <tns:ftpServer> element only: passiveMode

The following attributes are valid for the <tns:ftpsServer> element only: ftpsType, trustStore, trustStorePassword, trustStoreType, keyStore, keyStorePassword, keyStoreType, ccc, protFirst, auth, and connectTimeout.

The following attributes are valid for the <tns:limits> element within all of the <tns:ftpServer>, <tns:ftpsServer> and <tns:sftpServer> elements: maxListFileNames, maxListDirectoryLevels, maxReconnectRetry, reconnectWaitPeriod, maxSessions and socketTimeout

```
-->
<tns:serverProperties xmlns:tns="http://wmsqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmsqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">

  <!-- By default the location of the credentials file is in the home directory of the user
  that started the -->
  <!-- protocol bridge agent. If you wish to specify a different location use the
  credentialsFile element to -->
  <!-- do this. For
  example:
  <!-- <tns:credentialsFile path="/test/
  ProtocolBridgeCredentials.xml"/>
  <tns:defaultServer name="WINMVSCA.HURSLEY.IBM.COM" />
  <tns:ftpServer name="WINMVSCA.HURSLEY.IBM.COM" host="WINMVSCA.HURSLEY.IBM.COM"
  platform="unix"
  timeZone="Europe/London" locale="en-GB" fileEncoding="US-ASCII"
  listFormat="unix" limitedWrite="false" />

  <!-- Define servers here -->
</tns:serverProperties>
```

La commande peut générer le message suivant: BFGCL0532I:

Cet agent ne peut fonctionner que si un fichier de données d'identification supplémentaire est créé manuellement. Par défaut, ce fichier est appelé ProtocolBridgeCredentials.xml et se trouve dans le répertoire principal répertoire de l'utilisateur qui démarre l'agent. Par exemple, si cet utilisateur a démarré l'agent l'emplacement serait: \$HOME/ProtocolBridgeCredentials.xml

Si vous utilisez un fichier de données d'identification:

1. Pour plus d'informations sur la façon d'en créer un, voir le texte suivant.
2. Le fichier de données d'identification doit se trouver dans un répertoire avec des droits d'accès restreints. Par exemple, il ne doit pas y avoir d'accès en lecture pour les autres utilisateurs.
3. Indiquez l'emplacement du répertoire du fichier de données d'identification dans la variable d'environnement \$HOME pour l'ID utilisateur de l'agent démarré, ou éditez le fichier ProtocolBridgeProperties.xml et indiquez l'emplacement dans:

```
<tns:credentialsFile path="/test/ProtocolBridgeCredentials.xml"/>
```

Si vous souhaitez ajouter d'autres serveurs de protocole autres que ceux par défaut, éditez ce fichier pour définir leurs propriétés. Cet exemple ajoute un serveur FTP supplémentaire.

**Remarque :** L'agent de pont de protocole ne prend pas en charge le verrouillage de fichier. En effet, Managed File Transfer ne prend pas en charge le mécanisme de verrouillage de fichier sur un serveur de fichiers.

## Procédure

1. Définissez un serveur de fichiers de protocole en insérant les lignes suivantes dans le fichier en tant qu'élément enfant de <tns:serverProperties>:

```
<tns:ftpServer name="myserver" host="myhost.hursley.ibm.com" port="1234"
platform="windows"
        timeZone="Europe/London" locale="en-GB" fileEncoding="UTF-8"
        listFormat="unix" limitedWrite="false" >
<tns:limits maxListFileNames="10" maxListDirectoryLevels="500"/>
```

2. Modifiez ensuite la valeur des attributs:

- name est le nom de votre serveur de fichiers de protocole
- host est le nom d'hôte ou l'adresse IP du serveur de fichiers de protocole
- port est le numéro de port du serveur de fichiers de protocole
- platform est la plateforme sur laquelle s'exécute le serveur de fichiers de protocole
- timeZone est le fuseau horaire dans lequel s'exécute le serveur de fichiers de protocole
- locale est la langue utilisée sur le serveur de fichiers de protocole
- fileEncoding est le codage de caractères du serveur de fichiers de protocole
- listFormat est le format de liste de fichiers renvoyé par le serveur de fichiers de protocole
- limitedWrite détermine s'il convient de suivre le mode par défaut lors de l'écriture sur un serveur de fichiers, qui consiste à créer un fichier temporaire, puis à renommer ce fichier lorsque le transfert est terminé. Pour un serveur de fichiers configuré en écriture seule, le fichier est créé directement avec son nom final. La valeur de cette propriété peut être true ou false. L'attribut limitedWrite et la propriété d'agent doNotUseTempOutputFile sont utilisés ensemble dans le cas des agents de pont de protocole. Si vous souhaitez utiliser des fichiers temporaires, vous ne devez pas définir la valeur de doNotUseTempOutputFile et vous devez définir la valeur de limitedWrite sur false. Toute autre combinaison de paramètres signifie que les fichiers temporaires ne seront pas utilisés.
- maxListFileNames est le nombre maximal de noms collectés lors de l'analyse d'un répertoire sur le serveur de fichiers de protocole à la recherche de noms de fichier.
- maxListDirectoryLevels est le nombre maximal de niveaux de répertoire à récursivité lors de l'analyse d'un répertoire sur le serveur de fichiers de protocole à la recherche de noms de fichier.

Pour plus de détails sur ces attributs, notamment pour savoir s'ils sont obligatoires ou facultatifs et leurs valeurs par défaut, voir [Format de fichier de propriétés de pont de protocole](#).

### Référence associée

[Format du fichier de propriétés du pont de protocole](#)

[Expressions régulières utilisées par MFT](#)

## Recherche des propriétés du serveur de fichiers de protocole: ProtocolBridgePropertiesExit2

Si vous disposez d'un grand nombre de serveurs de fichiers de protocole, vous pouvez implémenter l'interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2` pour rechercher les propriétés de serveur de fichiers de protocole référencées dans les transferts. Vous pouvez implémenter cette interface de préférence à la gestion d'un fichier `ProtocolBridgeProperties.xml`.

### Pourquoi et quand exécuter cette tâche

Managed File Transfer fournit un exemple d'exit utilisateur qui recherche les propriétés du serveur de fichiers de protocole. Pour plus d'informations, voir [«Utilisation de l'exemple d'exit utilisateur pour rechercher des propriétés de serveur de fichiers de protocole»](#), à la page 308.

Tout exit utilisateur qui recherche des propriétés de pont de protocole doit implémenter l'interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit2`. Pour plus d'informations, voir [ProtocolBridgePropertiesExit.java](#) interface.

Vous pouvez chaîner plusieurs exits de propriétés de serveur de protocole de la même manière que d'autres exits utilisateur. Les exits sont appelés dans l'ordre dans lequel ils sont spécifiés à l'aide de la propriété `protocolBridgePropertiesExitClasses` dans le fichier de propriétés de l'agent. Les méthodes d'initialisation sont toutes renvoyées séparément et si une ou plusieurs d'entre elles renvoient la valeur `false`, l'agent ne démarre pas. L'erreur est signalée dans le journal des événements de l'agent.

Un seul résultat global est renvoyé pour les méthodes `getProtocolServerProperties` de tous les exits. Si la méthode renvoie un objet de propriétés comme code de résultat, cette valeur correspond au résultat renvoyé et les méthodes `getProtocolServerProperties` des exits suivants ne sont pas appelées. Si la méthode renvoie une valeur `null` comme code de résultat, la méthode `getProtocolServerProperties` de l'exit suivant est appelée. S'il n'y a pas d'exit suivant, le résultat `null` est renvoyé. Un code de résultat global nul est considéré comme un échec de recherche par l'agent de pont de protocole.

Il est recommandé d'utiliser l'interface `ProtocolBridgePropertiesExit2.java`, mais pour plus d'informations sur l'interface `ProtocolBridgePropertiesExit.java`, voir «Recherche des propriétés du serveur de fichiers de protocole: `ProtocolBridgePropertiesExit`», à la page 309.

Pour exécuter votre exit, procédez comme suit:

## Procédure

1. Compilez l'exit utilisateur des propriétés du serveur de protocole.
2. Créez un fichier d'archive Java (JAR) contenant l'exit compilé et sa structure de package.
3. Placez le fichier JAR contenant la classe d'exit dans le répertoire `exits` de l'agent de pont de protocole. Ce répertoire se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.
4. Editez le fichier de propriétés de l'agent de pont de protocole pour inclure la propriété `protocolBridgePropertiesExitClasses`. Pour la valeur de cette propriété, spécifiez une liste de classes séparées par des virgules qui implémentent un exit utilisateur de propriétés de serveur de pont de protocole. Les classes d'exit sont appelées dans l'ordre dans lequel elles sont spécifiées dans cette liste. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).
5. Vous pouvez éventuellement spécifier la propriété `protocolBridgePropertiesConfiguration`. La valeur que vous spécifiez pour cette propriété est transmise en tant que chaîne à la méthode `initialize()` des classes d'exit spécifiées par `protocolBridgePropertiesExitClasses`. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).

## **Utilisation de l'exemple d'exit utilisateur pour rechercher des propriétés de serveur de fichiers de protocole**

Managed File Transfer fournit un exemple d'exit utilisateur qui recherche les propriétés du serveur de fichiers de protocole.

## Pourquoi et quand exécuter cette tâche

Un exemple d'exit utilisateur qui recherche les propriétés de pont de protocole est fourni dans le répertoire `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` et dans la rubrique [Exemple d'exit utilisateur des propriétés de pont de protocole](#).

L'exit `SamplePropertiesExit2.java` lit un fichier de propriétés qui contient les propriétés des serveurs de protocole. Le format de chaque entrée du fichier de propriétés est le suivant:

```
serverName=type://host:port
```

L'emplacement du fichier de propriétés est extrait de la propriété d'agent de pont de protocole `protocolBridgePropertiesConfiguration`.

Pour exécuter l'exemple d'exit utilisateur, procédez comme suit:

## Procédure

1. Compilez le fichier `SamplePropertiesExit2.java`.
2. Créez un fichier JAR contenant l'exit compilé et sa structure de package.
3. Placez le fichier JAR dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/exits`.
4. Editez le fichier `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` pour qu'il contienne la ligne suivante:

```
protocolBridgePropertiesExitClasses=SamplePropertiesExit2
```

5. Créez un fichier de propriétés de pont de protocole, par exemple `protocol_bridge_properties.properties`, dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent`. Modifiez ce fichier pour inclure des entrées au format suivant:

```
serverName=type://host:port
```

6. Editez le fichier `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/agent.properties` pour qu'il contienne la ligne suivante:

```
protocolBridgePropertiesConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent/protocol_bridge_properties.properties
```

Vous devez utiliser le chemin d'accès absolu au fichier `protocol_bridge_properties.properties`.

7. Démarrez l'agent de pont de protocole à l'aide de la commande **fteStartAgent**.

## Concepts associés

«Le pont de protocole», à la page 302

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT, dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

## Référence associée

[Interface ProtocolBridgePropertiesExit.java](#)

[Exemple d'exit utilisateur de propriétés de pont de protocole](#)

[Le fichier MFT agent.properties](#)

[fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT\)](#)

## Recherche des propriétés du serveur de fichiers de protocole:

### ProtocolBridgePropertiesExit

Si vous disposez d'un grand nombre de serveurs de fichiers de protocole, vous pouvez implémenter l'interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` pour rechercher les propriétés de serveur de fichiers de protocole référencées dans les transferts.

## Pourquoi et quand exécuter cette tâche

Vous pouvez implémenter l'interface

`com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit` de préférence à la gestion d'un fichier `ProtocolBridgeProperties.xml`. Utilisez l'interface `ProtocolBridgePropertiesExit2.java`. La méthode **getCredentialLocation** dans `ProtocolBridgePropertiesExit2.java` utilise l'emplacement par défaut du fichier `ProtocolBridgeCredentials.xml`, qui correspond à votre répertoire de base.

Tout exit utilisateur qui recherche des propriétés de pont de protocole doit implémenter l'interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgePropertiesExit`:

```
package com.ibm.wmqfte.exitroutine.api;

import java.util.Map;
import java.util.Properties;

/**
 * An interface that is implemented by classes that are to be invoked as part of
 * user exit routine processing. This interface defines methods that will be
 * invoked by a protocol bridge agent to look up properties for protocol servers
 * that are referenced in transfers.
 * <p>
 * There will be one instance of each implementation class for each protocol
 * bridge agent. The methods can be called from different threads so the methods
 * must be synchronised.
 */
public interface ProtocolBridgePropertiesExit {

    /**
     * Invoked once when a protocol bridge agent is started. It is intended to
     * initialize any resources that are required by the exit.
     *
     * @param bridgeProperties
     *     The values of properties defined for the protocol bridge.
     *     These values can only be read, they cannot be updated by the
     *     implementation.
     * @return {@code true} if the initialization is successful and {@code
     *     false} if unsuccessful. If {@code false} is returned from an exit
     *     the protocol bridge agent will not start.
     */
    public boolean initialize(final Map<String, String> bridgeProperties);

    /**
     * Obtains a set of properties for the specified protocol server name.
     * <p>
     * The returned {@link Properties} must contain entries with key names
     * corresponding to the constants defined in
     * {@link ProtocolServerPropertyConstants} and in particular must include an
     * entry for all appropriate constants described as required.
     *
     * @param protocolServerName
     *     The name of the protocol server whose properties are to be
     *     returned. If a null or a blank value is specified, properties
     *     for the default protocol server are to be returned.
     * @return The {@link Properties} for the specified protocol server, or null
     *     if the server cannot be found.
     */
    public Properties getProtocolServerProperties(
        final String protocolServerName);

    /**
     * Invoked once when a protocol bridge agent is shut down. It is intended to
     * release any resources that were allocated by the exit.
     *
     * @param bridgeProperties
     *     The values of properties defined for the protocol bridge.
     *     These values can only be read, they cannot be updated by the
     *     implementation.
     */
    public void shutdown(final Map<String, String> bridgeProperties);
}
}
```

Vous pouvez chaîner plusieurs exits de propriétés de serveur de protocole de la même manière que d'autres exits utilisateur. Les exits sont appelés dans l'ordre dans lequel ils sont spécifiés à l'aide de la propriété `protocolBridgePropertiesExitClasses` dans le fichier de propriétés de l'agent. Les méthodes d'initialisation sont toutes renvoyées séparément et si une ou plusieurs d'entre elles renvoient la valeur `false`, l'agent ne démarre pas. L'erreur est signalée dans le journal des événements de l'agent.

Un seul résultat global est renvoyé pour les méthodes `getProtocolServerProperties` de tous les exits. Si la méthode renvoie un objet de propriétés comme code de résultat, cette valeur correspond au résultat renvoyé et les méthodes `getProtocolServerProperties` des exits suivants ne sont pas appelées. Si la méthode renvoie une valeur `null` comme code de résultat, la méthode `getProtocolServerProperties` de

l'exit suivant est appelée. S'il n'y a pas d'exit suivant, le résultat null est renvoyé. Un code de résultat global nul est considéré comme un échec de recherche par l'agent de pont de protocole.

## Procédure

Pour exécuter votre exit, procédez comme suit:

1. Compilez l'exit utilisateur des propriétés du serveur de protocole.
2. Créez un fichier d'archive Java (JAR) contenant l'exit compilé et sa structure de package.
3. Placez le fichier JAR contenant la classe d'exit dans le répertoire exits de l'agent de pont de protocole.

Ce répertoire se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.

4. Editez le fichier de propriétés de l'agent de pont de protocole pour inclure la propriété `protocolBridgePropertiesExitClasses`.

Pour la valeur de cette propriété, spécifiez une liste de classes séparées par des virgules qui implémentent un exit utilisateur de propriétés de serveur de pont de protocole. Les classes d'exit sont appelées dans l'ordre dans lequel elles sont spécifiées dans cette liste. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).

5. Vous pouvez éventuellement spécifier la propriété `protocolBridgePropertiesConfiguration`.

La valeur que vous spécifiez pour cette propriété est transmise en tant que chaîne à la méthode `initialize()` des classes d'exit spécifiées par `protocolBridgePropertiesExitClasses`. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).

## Mappage des données d'identification pour un serveur de fichiers

Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole ou en écrivant votre propre exit utilisateur. Managed File Transfer fournit un exemple d'exit utilisateur qui effectue le mappage des données d'identification de l'utilisateur.

### Concepts associés

[«Le pont de protocole», à la page 302](#)

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT, dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

### Tâches associées

[«Mappage des données d'identification pour un serveur de fichiers à l'aide du fichier ProtocolBridgeCredentials.xml», à la page 312](#)

Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole. Managed File Transfer fournit un fichier XML que vous pouvez éditer pour inclure vos données d'identification.

[«Mappage des données d'identification pour un serveur de fichiers à l'aide de classes d'exit», à la page 314](#)

Si vous ne souhaitez pas utiliser la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole, vous pouvez mapper les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers en écrivant votre propre exit utilisateur. Si vous configurez des exits utilisateur de mappage de données d'identification, ils prennent la place de la fonction de mappage de données d'identification par défaut.

[«Exemple: Comment configurer un agent de pont de protocole pour utiliser des données d'identification de clé privée avec un serveur SFTP UNIX», à la page 316](#)

Cet exemple explique comment générer et configurer le fichier `ProtocolBridgeCredentials.xml`. Cet exemple est typique et les détails peuvent varier en fonction de votre plateforme, mais les principes restent les mêmes.

### Référence associée

[Interface ProtocolBridgeCredentialExit.java](#)

[Exemple d'exit utilisateur de données d'identification de pont de protocole](#)

[Le fichier MFT `agent.properties`](#)

### **Mappage des données d'identification pour un serveur de fichiers à l'aide du fichier `ProtocolBridgeCredentials.xml`**

Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole. Managed File Transfer fournit un fichier XML que vous pouvez éditer pour inclure vos données d'identification.

### Pourquoi et quand exécuter cette tâche

Le fichier `ProtocolBridgeCredentials.xml` doit être créé manuellement par l'utilisateur. Par défaut, l'emplacement de ce fichier est le répertoire de base de l'utilisateur qui a démarré l'agent de pont de protocole, mais il peut être stocké n'importe où sur le système de fichiers accessible par l'agent. Pour spécifier un autre emplacement, ajoutez l'élément `<credentialsFile>` au fichier `ProtocolBridgeProperties.xml`. Exemple :

```
<tns:credentialsFile path="/example/path/to/ProtocolBridgeCredentials.xml"/>
```

Avant de pouvoir utiliser un agent de pont de protocole, configurez le mappage des données d'identification en éditant ce fichier pour inclure les informations d'hôte, d'utilisateur et de données d'identification. Pour plus d'informations et des exemples, voir [Format de fichier des données d'identification du pont de protocole](#).

### Procédure

1. Editez la ligne `<tns:server name="server name">` pour remplacer la valeur de l'attribut de nom par le nom du serveur dans le fichier `ProtocolBridgeProperties.xml`.

Vous pouvez utiliser l'attribut de modèle pour indiquer que vous avez utilisé un nom de serveur contenant des caractères génériques ou des expressions régulières. Exemple :

```
<tns:server name="serverA*" pattern="wildcard">
```

2. Insérez l'ID utilisateur et les données d'identification dans le fichier en tant qu'éléments enfant de `<tns:server>`.

Vous pouvez insérer un ou plusieurs des éléments suivants dans le fichier:

- Si le serveur de fichiers de protocole est un serveur FTP, FTPS ou SFTP, vous pouvez utiliser des mots de passe pour authentifier l'utilisateur qui demande le transfert. Insérez les lignes suivantes dans le fichier:

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  serverPassword="Server Password">
</tns:user>
```

Modifiez ensuite la valeur des attributs.

- `name` est une expression régulière Java qui correspond à l'ID utilisateur MQMD associé à la demande de transfert MFT



- serverUserId est la valeur qui est transmise au serveur de fichiers de protocole en tant qu'ID utilisateur de connexion. Si l'attribut serverUserId n'est pas spécifié, l'ID utilisateur MQMD associé à la demande de transfert MFT est utilisé à la place.
- serverPassword est le mot de passe associé à serverUserId.

L'attribut name peut contenir une expression régulière Java . Le programme de mappage de données d'identification tente de faire correspondre l'ID utilisateur MQMD de la demande de transfert MFT à cette expression régulière. L'agent de pont de protocole tente de faire correspondre l'ID utilisateur MQMD à l'expression régulière dans l'attribut de nom des éléments < tns: user> dans l'ordre dans lequel les éléments existent dans le fichier. Lorsqu'une correspondance est trouvée, l'agent de pont de protocole ne recherche pas d'autres correspondances. Si une correspondance est trouvée, les valeurs serverUserId et serverPassword correspondantes sont transmises au serveur de fichiers de protocole en tant qu'ID utilisateur et mot de passe de connexion. Les correspondances d'ID utilisateur MQMD sont sensibles à la casse.

- Si le serveur de fichiers de protocole est un serveur SFTP, vous pouvez utiliser des clés publiques et privées pour authentifier l'utilisateur qui demande le transfert. Insérez les lignes suivantes dans le fichier et modifiez la valeur des attributs. L'élément < tns: user> peut contenir un ou plusieurs éléments < tns:privateKey>.

```
<tns:user name="FTE User ID"
  serverUserId="Server User ID"
  hostKey="Host Key">
  <tns:privateKey associationName="association"
    keyPassword="Private key password">
    Private key file text
  </tns:privateKey>
</tns:user>
```

- name est une expression régulière Java qui correspond à l'ID utilisateur MQMD associé à la demande de transfert MFT
- serverUserId est la valeur qui est transmise au serveur de fichiers de protocole en tant qu'ID utilisateur de connexion. Si l'attribut serverUserId n'est pas spécifié, l'ID utilisateur MQMD associé à la demande de transfert MFT est utilisé à la place.
- hostKey est la clé attendue renvoyée par le serveur lors de la connexion
- key est la clé privée de serverUserId
- keyPassword est le mot de passe de la clé permettant de générer des clés publiques
- associationName est une valeur utilisée pour identifier à des fins de trace et de consignation

L'attribut name peut contenir une expression régulière Java . Le programme de mappage de données d'identification tente de faire correspondre l'ID utilisateur MQMD de la demande de transfert MFT à cette expression régulière. L'agent de pont de protocole tente de faire correspondre l'ID utilisateur MQMD à l'expression régulière dans l'attribut de nom des éléments < tns: user> dans l'ordre dans lequel les éléments existent dans le fichier. Lorsqu'une correspondance est trouvée, l'agent de pont de protocole ne recherche pas d'autres correspondances. Si une correspondance est trouvée, les valeurs serverUserId et key correspondantes sont utilisées pour authentifier l'utilisateur MFT auprès du serveur de fichiers de protocole. Les correspondances d'ID utilisateur MQMD sont sensibles à la casse.

Pour plus d'informations sur l'utilisation de clés privées avec un agent de pont de protocole, voir [«Exemple: Comment configurer un agent de pont de protocole pour utiliser des données d'identification de clé privée avec un serveur SFTP UNIX»](#), à la page 316.

**Remarque :** 

Lorsque la demande de transfert est écrite dans la file d'attente de commandes, l'ID utilisateur MQMD peut être converti en majuscules si la file d'attente de commandes de l'agent source se trouve sur un système z/OS ou IBM i . Par conséquent, l'ID utilisateur MQMD du même utilisateur d'origine peut arriver à l'exit de données d'identification dans la casse d'origine ou être converti en majuscules en fonction de l'agent source spécifié dans la demande de transfert. L'exit de mappage des données

d'identification par défaut effectue des correspondances sensibles à la casse avec l'ID utilisateur MQMD fourni, que vous devrez peut-être autoriser dans le fichier de mappage.

### Référence associée

[Format de fichier des données d'identification du pont de protocole](#)

[Format du fichier de propriétés du pont de protocole](#)

[Expressions régulières utilisées par MFT](#)

### ***Mappage des données d'identification pour un serveur de fichiers à l'aide de classes d'exit***

Si vous ne souhaitez pas utiliser la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole, vous pouvez mapper les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers en écrivant votre propre exit utilisateur. Si vous configurez des exits utilisateur de mappage de données d'identification, ils prennent la place de la fonction de mappage de données d'identification par défaut.

### **Pourquoi et quand exécuter cette tâche**

Managed File Transfer fournit un exemple d'exit utilisateur qui effectue le mappage des données d'identification de l'utilisateur. Pour plus d'informations, voir [«Utilisation de l'exemple d'exit utilisateur de données d'identification de pont de protocole»](#), à la page 315.

Un exit utilisateur pour le mappage des données d'identification de pont de protocole doit implémenter l'une des interfaces suivantes:

- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`, qui permet à un agent de pont de protocole de transférer des fichiers vers et depuis un serveur de fichiers de protocole par défaut
- `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2`, qui vous permet de transférer des fichiers vers et depuis plusieurs noeuds finaux.

L'interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit2` contient la même fonction que `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit` et inclut également une fonction étendue. Pour plus d'informations, voir [InterfaceProtocolBridgeCredentialExit.java](#) et [interfaceProtocolBridgeCredentialExit2.java](#).

Les exits de données d'identification peuvent être chaînés de la même manière que les autres exits utilisateur. Les exits sont appelés dans l'ordre dans lequel ils sont spécifiés à l'aide de la propriété `protocolBridgeCredentialConfiguration` dans le fichier de propriétés de l'agent. Les méthodes d'initialisation sont toutes renvoyées séparément et si une ou plusieurs d'entre elles renvoient la valeur `false`, l'agent ne démarre pas. L'erreur est signalée dans le journal des événements de l'agent.

Un seul résultat global est renvoyé pour les méthodes d'ID `mapMQUser` de tous les exits comme suit:

- Si la méthode renvoie la valeur `USER_SUCCESSFULLY_MAPPING` ou `USER_NIGED_ACCESS` comme code de résultat, cette valeur est le résultat renvoyé et les méthodes d'ID `mapMQUser` des exits suivants ne sont pas appelées.
- Si la méthode renvoie la valeur `NO_MAPPING_FOUND` comme code de résultat, la méthode d'ID `mapMQUser` de l'exit suivant est appelée.
- S'il n'y a pas d'exit suivant, le résultat `NO_MAPPING_FOUND` est renvoyé.
- Un code de résultat global de `USER_DÉNIED_ACCESS` ou `NO_MAPPING_FOUND` est considéré comme un échec de transfert par l'agent de pont.

Pour exécuter votre exit, procédez comme suit:

### **Procédure**

1. Compilez l'exit utilisateur des données d'identification du pont de protocole.
2. Créez un fichier d'archive Java (JAR) contenant l'exit compilé et sa structure de package.

- Placez le fichier JAR qui contient la classe d'exit dans le répertoire `exits` de l'agent de pont. Le répertoire se trouve dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name`.
- Editez le fichier de propriétés de l'agent de pont de protocole pour inclure la propriété `protocolBridgeCredentialExitClasses`. Pour la valeur de cette propriété, spécifiez une liste de classes séparées par des virgules qui implémentent une routine d'exit de données d'identification de pont de protocole. Les classes d'exit sont appelées dans l'ordre dans lequel elles sont spécifiées dans cette liste. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).
- Editez le fichier de propriétés de l'agent de pont de protocole pour inclure:

```
exitClassPath=IBM MQ
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Le fichier `agent.properties` d'un agent se trouve dans votre répertoire `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/bridge_agent_name`.

Si vous modifiez le fichier `agent.properties`, vous devez redémarrer l'agent pour appliquer les modifications.

- Vous pouvez éventuellement spécifier la propriété `protocolBridgeCredentialConfiguration`. La valeur que vous spécifiez pour cette propriété est transmise en tant qu'objet `String` à la méthode `initialize()` des classes d'exit spécifiées par `protocolBridgeCredentialExitClasses`. Pour plus d'informations, voir [Le fichier MFT agent.properties](#).
- Démarrez l'agent de pont de protocole à l'aide de la commande **`fteStartAgent`**.

*Utilisation de l'exemple d'exit utilisateur de données d'identification de pont de protocole*  
Managed File Transfer fournit un exemple d'exit utilisateur qui effectue le mappage des données d'identification de l'utilisateur.

## Pourquoi et quand exécuter cette tâche

Un exemple d'exit de données d'identification de pont de protocole est fourni dans le répertoire `MQ_INSTALLATION_PATH/mqft/samples/protocolBridge` et dans la rubrique [Exemple d'exit utilisateur de données d'identification de pont de protocole](#). Cet exemple est basé sur l'interface `com.ibm.wmqfte.exitroutine.api.ProtocolBridgeCredentialExit`.

L'exit `SampleCredentialExit.java` lit un fichier de propriétés qui mappe les ID utilisateur MQMD associés aux demandes de transfert vers les ID utilisateur du serveur et les mots de passe du serveur. L'emplacement du fichier de propriétés est extrait de la propriété d'agent de pont de protocole `protocolBridgeCredentialConfiguration`.

Pour exécuter l'exemple d'exit utilisateur, procédez comme suit:

## Procédure

- Compilez le fichier `SampleCredentialExit.java`.
- Créez un fichier JAR contenant l'exit compilé et sa structure de package.
- Placez le fichier JAR dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/exits`.
- Editez le fichier `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` pour qu'il contienne la ligne suivante:

```
protocolBridgeCredentialExitClasses=SampleCredentialExit
```

- Editez le fichier de propriétés de l'agent de pont de protocole pour inclure:

```
exitClassPath=IBM MQ
```

```
installation_directory\mqft\config\configuration_queue_manager\agents\protocol_bridge_agent_name\exits\SampleCredentialExit.jar
```

Le fichier agent.properties d'un agent se trouve dans votre répertoire `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name`.

Si vous modifiez le fichier agent.properties, vous devez redémarrer l'agent pour appliquer les modifications.

6. Créez un fichier de propriétés de données d'identification (credentials.properties) dans le répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent` et éditez-le pour inclure des entrées au format suivant:

```
mqUserId=serverUserId,serverPassword
```

7. Editez le fichier `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/agent.properties` pour qu'il contienne la ligne suivante:

```
protocolBridgeCredentialConfiguration=MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/bridge_agent_name/credentials.properties
```

Vous devez utiliser le chemin d'accès absolu au fichier credentials.properties.

8. Démarrez l'agent de pont de protocole à l'aide de la commande **fteStartAgent**.

### Concepts associés

«Le pont de protocole», à la page 302

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT, dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

### Référence associée

[Interface ProtocolBridgeCredentialExit.java](#)

[Interface ProtocolBridgeCredentialExit2.java](#)

[Exemple d'exit utilisateur de données d'identification de pont de protocole](#)

[Le fichier MFT agent.properties](#)

[fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT\)](#)

## Exemple: Comment configurer un agent de pont de protocole pour utiliser des données d'identification de clé privée avec un serveur SFTP UNIX

Cet exemple explique comment générer et configurer le fichier ProtocolBridgeCredentials.xml. Cet exemple est typique et les détails peuvent varier en fonction de votre plateforme, mais les principes restent les mêmes.

## Pourquoi et quand exécuter cette tâche

### Procédure

1. Générez une clé publique et une clé privée à utiliser pour l'authentification auprès du serveur SFTP.

Par exemple, sur un système hôte Linux, vous pouvez utiliser l'outil **ssh-keygen**, fourni avec le package 'openssh', pour créer la paire clé publique / clé privée.

Par défaut, sans arguments, la commande **ssh-keygen** vous invite à indiquer un emplacement et une phrase passe pour les deux fichiers de clés, dont les noms par défaut sont les suivants:

```
id_rsa      <-- Private key
id_rsa.pub  <-- Public key
```



**Avertissement :** Si vous utilisez la commande **ssh-keygen** à partir d'une version récente d' OpenSSH, telle que celle fournie avec RHEL 8, le format de clé utilisé n'est pas compatible avec l'agent de pont de protocole et les tentatives de transfert vers le serveur SFTP échouent avec le message suivant:

```
BFGBR0216E: Authentication to protocol server 'sftp.host.address' failed
because of invalid private key.
```

Pour créer une clé privée compatible avec ces nouvelles versions d' OpenSSH, spécifiez le format de clé avec l'argument suivant dans la commande **ssh-keygen** :

```
ssh-keygen -m PEM
```

Le contenu de la clé privée `id_rsa` contient alors les première et dernière lignes de:

```
-----BEGIN RSA PRIVATE KEY-----
... ..
-----END RSA PRIVATE KEY-----
```

qui est compatible avec l'agent de pont de protocole.

2. Copiez l'intégralité du contenu du fichier `id_rsa.pub` dans le fichier `~/ .ssh/authorized_keys` de l'utilisateur SFTP sur le serveur SFTP.

Vérifiez que les droits d'accès à ce fichier et au répertoire `~/ .ssh` sont définis de manière appropriée pour que le serveur SFTP autorise l'authentification par clé. Ces droits sont généralement les suivants:

```
~/ .ssh                Mode 700
~/ .ssh/authorized_keys Mode 600
```

3. Managed File Transfer requiert une empreinte digitale SSH d'hôte générée à l'aide de l'algorithme MD5 . Exécutez l'une des commandes suivantes pour obtenir l'empreinte SSL de l'hôte du serveur SFTP.
  - Pour Red Hat® Enterprise Linux version 6.x et les versions suivantes, et Linux Ubuntu 14.04, exécutez la commande suivante:

```
ssh-keygen -l -f /etc/ssh/ssh_host_rsa_key.pub
```

- A partir de Red Hat Enterprise Linux 7.x, Linux Ubuntu 16.04 et SuSE Linux 12.4, la commande `ssh-keygen` génère, par défaut, l'empreinte digitale ssh à l'aide de l'algorithme SHA256 . Pour générer l'empreinte digitale ssh à l'aide de l'algorithme MD5 , exécutez la commande suivante:

```
ssh-keygen -l -E MD5 -f /etc/ssh/ssh_host_rsa_key.pub
```

La sortie de la commande est similaire à l'exemple suivant:

```
2048 MD5:64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27 no comment (RSA)
```

Extrayez la partie hexadécimale uniquement de la sortie à utiliser comme `hostKey` dans le fichier `ProtocolBridgeCredentials.xml` (voir l'étape «4», à la page 317). Par conséquent, dans cet exemple, vous extrayez `64:39:f5:49:41:10:55:d2:0b:81:42:5c:87:62:9d:27`.

4. Sur le système d'agent de pont de protocole, éditez le fichier `ProtocolBridgeCredentials.xml` . Remplacez les valeurs affichées en italique dans l'exemple suivant par vos propres valeurs:

```
<tns:credentials xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeCredentials"
xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeCredentials
ProtocolBridgeCredentials.xsd ">

<tns:agent name="Agent_name">

<tns:server name="SFTP_name">
```

```

<tns:user name="mq_User_ID" serverUserId="SFTP_user_ID"
  hostKey="ssh_host_finger">
<tns:privateKey associationName="name" keyPassword="pass_phrase">
  Complete contents of the id_rsa file including the entries
  -----BEGIN RSA PRIVATE KEY-----

  -----END RSA PRIVATE KEY-----
</tns:privateKey>
</tns:user>

</tns:server>
</tns:agent>
</tns:credentials>

```

où :

- *nom\_agent* est le nom de l'agent de pont de protocole.
- *SFTP\_host\_name* est le nom du serveur SFTP, comme indiqué dans le fichier `ProtocolBridgeProperties.xml`.
- *mq\_User\_ID* est l'ID utilisateur MQMD associé à la demande de transfert.
- *SFTP\_user\_ID* est l'ID utilisateur SFTP tel qu'il est utilisé à l'étape 2. Il s'agit de la valeur transmise au SFTP qui sert d'ID utilisateur de connexion.
- *ssh\_host\_finger* est l'empreinte digitale collectée à l'étape 3.
- *name* est un nom que vous pouvez spécifier pour être utilisé à des fins de trace et de consignation.
- *pass\_phrase* est la phrase passe que vous avez fournie dans `ssh-keygen` à l'étape 1.
- *Contenu complet du fichier id\_rsa* est le contenu complet du fichier `id_rsa` généré à l'étape 1. Pour éviter une erreur de connexion, veillez à inclure les deux entrées suivantes:

```

-----BEGIN RSA PRIVATE KEY-----
-----END RSA PRIVATE KEY-----

```

Vous pouvez ajouter des clés supplémentaires en dupliquant l'élément `<tns:privatekey>`.

5. Démarrez l'agent de pont de protocole si l'agent n'est pas déjà démarré. Sinon, l'agent de pont de protocole interroge régulièrement le fichier `ProtocolBridgeCredentials.xml` et récupère les modifications.

### Référence associée

Format de fichier des données d'identification du pont de protocole

[fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT\)](#)

[Le fichier MFT agent.properties](#)

«Mappage des données d'identification pour un serveur de fichiers», à la page 311

Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole ou en écrivant votre propre exit utilisateur. Managed File Transfer fournit un exemple d'exit utilisateur qui effectue le mappage des données d'identification de l'utilisateur.

## Configuration d'un pont de protocole pour un serveur FTPS

Configurez un serveur FTPS de la même manière que vous configurez un serveur FTP: créez un agent de pont pour le serveur, définissez les propriétés du serveur et mappez les données d'identification de l'utilisateur.

### Pourquoi et quand exécuter cette tâche

Pour configurer un serveur FTPS, procédez comme suit:

## Procédure

1. Créez un agent de pont de protocole pour le serveur FTPS à l'aide de la commande **fteCreateBridgeAgent**. Les paramètres applicables à FTP sont également applicables à FTPS, mais il existe également trois paramètres obligatoires spécifiques à FTPS:
  - a) Le paramètre **-bt**. Spécifiez FTPS comme valeur de ce paramètre.
  - b) Paramètre **-bts** du fichier de clés certifiées. La commande suppose que seule l'authentification du serveur est requise et que vous devez spécifier l'emplacement du fichier de clés certifiées.

La forme explicite du protocole FTPS est configurée par la commande **fteCreateBridgeAgent** par défaut, mais vous pouvez configurer la forme implicite en modifiant le fichier de propriétés du pont de protocole. Le pont de protocole se connecte toujours aux serveurs FTPS en mode passif.

Pour plus d'informations sur la commande **fteCreateBridgeAgent**, voir [fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT\)](#).

Si vous avez besoin d'instructions pour créer des fichiers de clés certifiées, consultez les informations relatives à l'outil de clés dans la [documentation de l'outil de clésOracle](#).

2. Définissez les propriétés du serveur FTPS dans un élément `<ftpsServer>` dans le fichier de propriétés du pont de protocole: `ProtocolBridgeProperties.xml`. Pour plus d'informations, voir la section «Définition des propriétés des serveurs de fichiers de protocole à l'aide du fichier `ProtocolBridgeProperties.xml`», à la page 304. Vous pouvez également activer l'authentification client en éditant le fichier de propriétés du pont de protocole. Pour plus de détails sur toutes les options de configuration, voir [Format de fichier de propriétés de pont de protocole](#).
3. Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur FTPS à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole ou en écrivant votre propre exit utilisateur. Pour plus d'informations, voir «[Mappage des données d'identification pour un serveur de fichiers](#)», à la page 311.
4. Par défaut, le fichier de clés certifiées est configuré au format JKS ; si vous souhaitez modifier le format, éditez le fichier de propriétés du pont de protocole.

## Exemple

Voici un exemple d'entrée pour un serveur FTPS dans le fichier de propriétés du pont de protocole:

```
<tns:serverProperties xmlns:tns="http://wmqfte.ibm.com/ProtocolBridgeProperties"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/ProtocolBridgeProperties
  ProtocolBridgeProperties.xsd">
  <tns:defaultServer name="ftpsserver.mycompany.com" />

  <tns:ftpsServer name="ftpsserver.mycompany.com" host="ftpsserver.mycompany.com" port="990"
  platform="windows"
  timeZone="Europe/London" locale="en_US" fileEncoding="UTF8"
  listFormat="unix" limitedWrite="false"
  trustStore="c:\mydirec\truststore.jks" />

  <!-- Define servers here -->
</tns:serverProperties>
```

## Que faire ensuite

Pour plus d'informations sur les parties du protocole FTPS prises en charge et non prises en charge, voir [Prise en charge du serveur FTPS par le pont de protocole](#).

## Concepts associés

«Le pont de protocole», à la page 302

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT, dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur

de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

### Tâches associées

«Mappage des données d'identification pour un serveur de fichiers à l'aide du fichier ProtocolBridgeCredentials.xml», à la page 312

Mappez les données d'identification de l'utilisateur dans Managed File Transfer aux données d'identification de l'utilisateur sur le serveur de fichiers à l'aide de la fonction de mappage des données d'identification par défaut de l'agent de pont de protocole. Managed File Transfer fournit un fichier XML que vous pouvez éditer pour inclure vos données d'identification.

«Définition des propriétés des serveurs de fichiers de protocole à l'aide du fichier ProtocolBridgeProperties.xml», à la page 304

Définissez les propriétés d'un ou de plusieurs serveurs de fichiers de protocole vers et depuis lesquels vous souhaitez transférer des fichiers à l'aide du fichier ProtocolBridgeProperties.xml, fourni par Managed File Transfer dans le répertoire de configuration de l'agent.

### Référence associée

[fteCreateBridgeAgent \(créer et configurer un agent de pont de protocole MFT\)](#)

[Format de fichier des données d'identification du pont de protocole](#)

[Format du fichier de propriétés du pont de protocole](#)

[Prise en charge du serveur FTPS par le pont de protocole](#)

## Scénarios et exemples de limitation du nombre de transferts de fichiers vers des serveurs de fichiers individuels

Comment l'agent de pont de protocole révisé fonctionne avec les attributs

**maxActiveDestinationTransfers** et **failTransferWhenCapacityReached**, avec quelques exemples.

## Scénarios illustrant le fonctionnement de l'agent de pont de protocole en fonction de la valeur maxActiveDestinationTransfers

### Scénario 1

Le fichier ProtocolBridgeProperties.xml d'un agent de pont de protocole contient deux définitions de serveur de fichiers:

- Vous n'avez pas défini l'attribut global **maxActiveDestinationTransfers**.
- Vous n'avez pas défini l'attribut **maxActiveDestinationTransfers** à la fois sur fileServerA et sur FileServerB.
- Vous avez défini l'attribut **maxDestinationTransfers** de l'agent de pont de protocole sur la valeur par défaut.

Si vous avez défini l'attribut **maxDestinationTransfers** de l'agent de pont de protocole sur la valeur par défaut 25, alors:

- L'agent de destination démarre le traitement de deux transferts gérés vers fileServerA.
- Les deux transferts sont terminés.

A ce stade, le client se rend compte que fileServerA a échoué et définit les valeurs suivantes pour fileServerA dans le fichier ProtocolBridgeProperties.xml :

**maxActiveDestinationTransfers = 0**

**failTransferWhenCapacityReached = true**

- Un autre transfert arrive pour fileServerA et quelques-uns pour fileServerB:



En fonction des propriétés définies à l'étape précédente, le transfert géré vers `fileServerA` est rejeté et marqué comme ayant échoué, tandis que les transferts pour `fileServerB` sont gérés dans le flux existant standard.

- Après un certain temps, le client découvre que le `fileServerA` est à nouveau en cours d'exécution, de sorte que le client supprime ou met en commentaire la valeur ajoutée précédemment dans `ProtocolBridgeProperties.xml`. Un nouveau transfert géré arrive pour `fileServerA` et est géré dans le flux existant standard.

## Scénario 2

- Vous avez défini l'attribut **`maxActiveDestinationTransfers`** pour un serveur de fichiers et non l'attribut **`failTransferWhenCapacityReached`**.
- L'agent de pont de protocole sert d'agent de destination pour ce nombre de transferts gérés vers le serveur de fichiers.
- La valeur de l'attribut **`maxActiveDestinationTransfers`** est réduite de 1.

L'agent de pont de protocole met à jour dynamiquement sa configuration et définit **`maxActiveDestinationTransfers`** sur la nouvelle valeur alors qu'il est encore actif. Les transferts gérés en cours ne sont pas affectés par cette mise à jour et peuvent être terminés.

## Scénario 3

Le fichier `ProtocolBridgeProperties.xml` d'un agent de pont de protocole contient deux définitions de serveur de fichiers:

- Vous n'avez pas défini l'attribut global **`maxActiveDestinationTransfers`**.
- Vous n'avez pas défini l'attribut **`failTransferWhenCapacityReached`**.
- Vous avez défini **`maxActiveDestinationTransfers`** sur 1 sous `fileServerA`.
- Vous n'avez pas défini l'attribut **`maxActiveDestinationTransfers`** sur `fileServerB`.

Si l'attribut **`maxDestinationTransfers`** de l'agent de pont de protocole est défini sur 5:

- Le nombre maximal de transferts de destination actifs de l'agent de pont de protocole vers `fileServerA` est 1 (bien que l'agent de destination dispose de 5 emplacements de transfert de destination, un seul peut être utilisé pour les transferts gérés vers `fileServerA`).

Cela est utile en cas d'échec de `fileServerA`. Une fois que `fileServerA` est à nouveau en cours d'exécution, la valeur de **`maxActiveDestinationTransfers`** peut être augmentée à 5 pour permettre la pleine capacité des transferts de destination autorisés.

- Le nombre maximal de transferts de destination actifs de l'agent de pont de protocole vers `fileServerB` est 5.

Comme **`maxActiveDestinationTransfers`** n'est pas défini pour ce serveur de fichiers, l'agent de pont de protocole peut utiliser les 5 emplacements de transfert de destination pour les transferts gérés vers ce serveur.

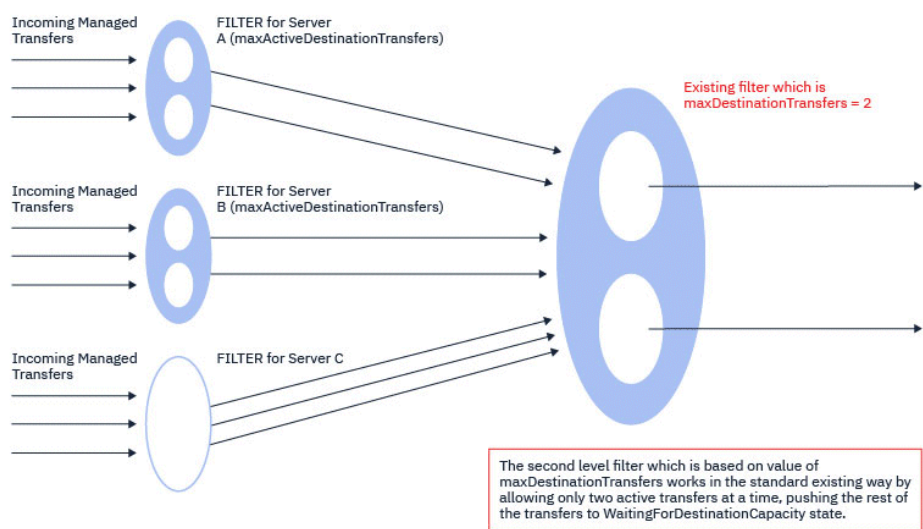
## Scénario 4

Dans le diagramme suivant:

- Vous avez défini l'attribut **`maxDestinationTransfers`** sur 2 dans le fichier `agent.properties`.
- Vous avez défini **`maxActiveDestinationTransfers`** sur 2 sous `fileServerA`.
- Vous avez défini l'attribut **`maxActiveDestinationTransfers`** sur 2 sous `fileServerB`.
- Vous n'avez pas défini l'attribut **`maxActiveDestinationTransfers`** sur `fileServerC`.

**SCENARIO**  
 maxDestinationTransfers=2  
 fileServerA :  
 maxActiveDestinationTransfers=2  
 fileServerB :  
 maxDestinationTransfers=2  
 fileServerC :  
 maxDestinationTransfers not set

In both cases A and B, only two managed transfers are let in by the maxActiveDestinationTransfers filter. The rest of the transfers go into the WaitingForDestinationFileServerCapacity state.  
 In the case of C, the filter gives everything a passthrough since no value is set.



The second level filter which is based on value of maxDestinationTransfers works in the standard existing way by allowing only two active transfers at a time, pushing the rest of the transfers to WaitingForDestinationCapacity state.

Comme le montre le diagramme, les attributs **maxActiveDestinationTransfers** et **maxDestinationTransfers** sont indépendants les uns des autres.

Les valeurs de **maxActiveDestinationTransfers** pour chacun des serveurs sont vérifiées. En fonction de cette valeur, les transferts sont autorisés à continuer ou à passer à l'état **WaitingForDestinationFileServerCapacity**.

Les transferts autorisés passent ensuite par le flux de contrôle standard existant par rapport au **maxDestinationTransfers**.

### Scénario 5



**Avertissement :** Vous devez être prudent lorsque vous définissez les valeurs des attributs **maxActiveDestinationTransfers** car vous devez garder la valeur de l'attribut **maxDestinationTransfers** à l'esprit.

Si vous ne le faites pas, une situation décrite dans le texte suivant peut se produire:

- Vous n'avez pas défini de valeur pour l'attribut global **maxActiveDestinationTransfers**.
- Vous avez défini la valeur **maxDestinationTransfers= 2** dans le fichier `agent.properties`.
- Vous avez défini la valeur **maxActiveDestinationTransfers= 2** sur `fileServerA`.
- Vous n'avez pas défini de valeur pour **maxActiveDestinationTransfers** sur `fileServerB`.

Supposons que la séquence d'événements suivante se produise:

- L'agent de pont de protocole reçoit une demande de transfert d'un fichier vers `fileServerA`. L'agent de pont de protocole n'est pas en train de faire quoi que ce soit, il accepte donc cette demande de transfert géré.

Les créneaux de transfert se ressemblent maintenant:

- Transferts de destination: 1
- Transferts de destination pour `fileServerA`: 1
- Transferts de destination pour `fileServerB`: 0
- A présent, l'agent de pont de protocole reçoit une autre demande pour agir en tant qu'agent de destination pour un transfert géré impliquant `fileServerA`. Une fois de plus, il accepte cette demande et les créneaux de transfert ressemblent à ceci:

- Transferts de destination: 2
- Transferts de destination pour fileServerA: 2
- Transferts de destination pour fileServerB: 0

Les deux emplacements Destination Transfer de l'agent sont maintenant occupés et l'agent ne peut donc pas participer à d'autres transferts gérés tant que l'un des transferts vers fileServerA n'est pas terminé.

- Peu de temps après, fileServerA échoue, ce qui entraîne la reprise des deux transferts gérés. Les emplacements Destination Transfer utilisés par ces transferts gérés restent en cours d'utilisation pendant cette période.
- Ensuite, l'agent de pont de protocole reçoit une demande de transfert d'un fichier vers fileServerB. Il existe un espace pour ce transfert dans les emplacements Destination Transfers for fileServerB. Cependant, tous les emplacements Destination Transfer de l'agent sont utilisés et le transfert est donc placé dans le journal des éléments en attente afin de pouvoir être relancé ultérieurement.

Par conséquent, le transfert vers fileServerB est bloqué jusqu'à ce qu'au moins un des transferts vers fileServerA soit terminé et libère son emplacement Destination Transfer.

Pour éviter cette situation:

- Définissez la valeur de **maxActiveDestinationTransfers** sur les serveurs de fichiers pour qu'elle soit inférieure à la valeur de **maxDestinationTransfers**, de sorte que les emplacements libres restent disponibles.
- Ou répartissez équitablement la valeur de l'attribut **maxActiveDestinationTransfers** entre tous les serveurs de noeud final.

## Comportement de l'agent de pont de protocole basé sur les valeurs de l'attribut maxActiveDestinationTransfers

**Remarque :** Dans tous les cas d'erreur répertoriés dans le tableau suivant, si l'attribut **maxActiveDestinationTransfers** est défini sur une valeur non valide, l'agent de pont de protocole suppose que cet attribut n'est pas défini.

maxActiveDestinationTransfers	Exemple de valeur	Description
Non indiqué	Non indiqué	Les transferts se passent comme d'habitude. Il n'y a pas de limite au nombre de transferts pour le noeud final * ftp *.
Spécifiée	0	Aucun transfert autorisé vers ce noeud final * ftp * spécifique.
Valeur négative	-1	Erreur consignée dans output0.log La valeur -1 n'est pas valide pour un entier non négatif. L'agent de pont de protocole suppose que l'attribut n'est pas défini.
Valeur non entière	abc	Erreur consignée dans output0.log La valeur abc n'est pas valide pour un entier. L'agent de pont de protocole suppose que l'attribut n'est pas défini.
Vide	""	La valeur '' de l'attribut maxActiveDestinationTransfers n'est pas valide pour un entier non négatif.

<b>maxActiveDestinationTransfers</b>	<b>Exemple de valeur</b>	<b>Description</b>
Spécifiée	5	Autorise uniquement cinq transferts actifs à s'exécuter à tout moment pour ce noeud final * ftp *.  Les transferts excessifs sont relancés ou rejetés, en fonction de la valeur de l'attribut failTransferWhenCapacityatteint.

### Comportement de l'agent de pont de protocole pour la combinaison des attributs maxActiveDestinationTransfers et failTransferWhenCapacityReached

<b>failTransferWhenCapacityValeur atteinte</b>	<b>Valeur maxActiveDestinationTransfers</b>	<b>Résultat</b>
Faux	3	Trois transferts actifs sont autorisés vers ce serveur de noeud final. Tous les autres transferts sont relancés.
True	3	Trois transferts actifs sont autorisés vers ce serveur de noeud final. Tous les autres transferts sont rejetés et marqués comme ayant échoué.
Non indiqué	3	La valeur par défaut <b>false</b> pour failTransferWhenCapacityest prise en compte.  Il en résulte que trois transferts actifs sont autorisés vers ce serveur de noeud final. Tous les autres transferts sont relancés.
Valeurs autres que la valeur booléenne	Spécifiée	Erreur consignée dans output.log.  La valeur spécifiée pour failTransferWhenCapacityatteint n'est pas une valeur booléenne.  La valeur par défaut de failTransferWhenCapacityest prise en compte.

### Comportement de l'agent de pont de protocole pour la combinaison des attributs maxDestinationTransfers et failTransferWhenCapacityReached

<b>failTransferWhenCapacityValeur atteinte</b>	<b>Valeur de transferts maxDestination</b>	<b>Résultat</b>
True	10	Lorsque le nombre de transferts simultanés actifs atteint 10, le 11 <sup>ème</sup> transfert géré est échoué par l'agent de pont de protocole.

<b>failTransferWhenCapacityValeur atteinte</b>	<b>Valeur de transferts maxDestination</b>	<b>Résultat</b>
Faux	10	Comportement existant. Lorsque le nombre de transferts simultanés actifs atteint 10, le 11 <sup>ème</sup> transfert géré est mis en file d'attente en attendant qu'un emplacement soit libéré.
Non indiqué	10	Comportement existant

## Messages d'erreur

Message existant:

### **BFGS0082I**

Est consigné dans le fichier output0.log de l'agent source lorsque l'agent de pont de protocole rejette le transfert, alors que l'agent de pont de protocole exécute déjà le nombre maximal de transferts défini dans l'attribut **maxDestinationTransfers**.

Nouveaux messages:

### **BFGSS0085I**

Est consigné dans le fichier output0.log de l'agent source lorsque l'agent de pont de protocole rejette et relance un transfert géré,

### **BFGSS0086I**

Est consigné dans le fichier output0.log de l'agent source lorsque l'agent de pont de protocole rejette et relance un transfert géré et que l'élément de destination n'inclut pas le nom du serveur de fichiers

### **BFGSS0084E**

Est consigné dans le fichier Explorer et audit.xml lorsque l'agent de pont de protocole est rejeté pour avoir dépassé le nombre maximal de transferts simultanés spécifié dans l'attribut **maxActiveDestinationTransfers** et marque un transfert géré comme ayant échoué.

### **BFGSS0087E**

Est consigné dans l'explorateur et le fichier audit.xml lorsque l'agent de pont de protocole est rejeté pour avoir dépassé le nombre maximal de transferts cible spécifié dans l'attribut **maxActiveDestinationTransfers** et marque un transfert géré comme ayant échoué.

### **BFGSS0088W**

Est consigné dans le journal output0.log lorsque la valeur de l'attribut **maxActiveDestinationTransfers** dépasse la valeur de l'attribut **maxDestinationTransfers**.

### **BFGSS0089I**

Est consigné dans le fichier output0.log de l'agent de pont de protocole de destination, lorsqu'il fonctionne avec un agent source qui ne se trouve pas dans IBM MQ 9.3.0 ou version ultérieure.

## Concepts associés

«Le pont de protocole», à la page 302

Le pont de protocole permet à votre réseau Managed File Transfer (MFT) d'accéder aux fichiers stockés sur un serveur de fichiers en dehors de votre réseau MFT, dans votre domaine local ou à un emplacement distant. Ce serveur de fichiers peut utiliser les protocoles de réseau FTP, FTPS ou SFTP. Chaque serveur de fichiers a besoin d'au moins un agent dédié. L'agent dédié est appelé agent de pont de protocole. Un agent de pont peut interagir avec plusieurs serveurs de fichiers.

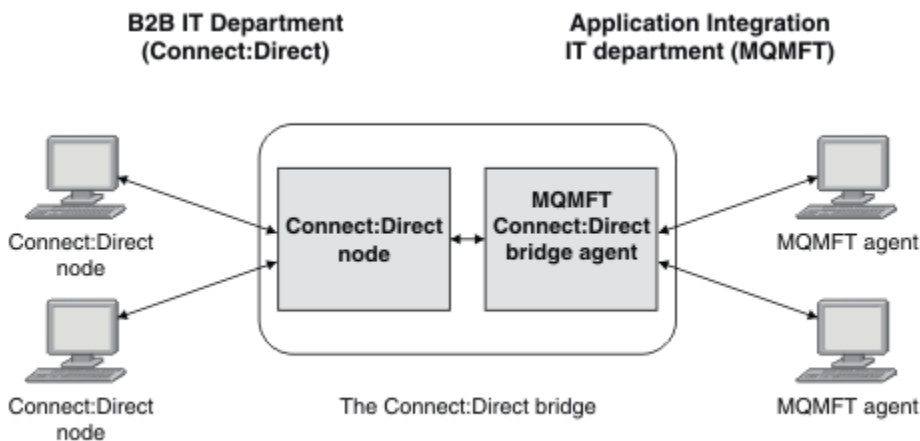
## Tâches associées

«Définition des propriétés des serveurs de fichiers de protocole à l'aide du fichier ProtocolBridgeProperties.xml», à la page 304

Définissez les propriétés d'un ou de plusieurs serveurs de fichiers de protocole vers et depuis lesquels vous souhaitez transférer des fichiers à l'aide du fichier ProtocolBridgeProperties.xml, fourni par Managed File Transfer dans le répertoire de configuration de l'agent.

## Le pont Connect:Direct

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.



Le diagramme montre un pont MFT Connect:Direct entre deux services, le service informatique B2B et le service informatique Application Integration. Le service informatique B2B utilise Connect:Direct pour transférer des fichiers depuis et vers les partenaires commerciaux de la société. Le service informatique d'intégration d'applications utilise IBM MQ comme infrastructure de messagerie et par conséquent, a récemment choisi Managed File Transfer comme solution de transfert de fichier.

A l'aide du pont MFT Connect:Direct, les deux services peuvent transférer des fichiers entre le réseau Connect:Direct du service informatique B2B et le réseau MFT du service informatique Application Integration. Le pont Connect:Direct est un composant de Managed File Transfer, qui inclut un agent MFT qui communique avec un noeud Connect:Direct. L'agent MFT est dédié aux transferts avec le noeud Connect:Direct : il s'agit de l'agent de pont Connect:Direct.

Le pont Connect:Direct est disponible dans les composants Service et Agent de Managed File Transfer et peut être utilisé pour les tâches suivantes :

1. Utiliser des commandes Managed File Transfer pour initier le transfert d'un fichier ou de plusieurs fichiers depuis un agent MFT vers un noeud Connect:Direct.
2. Utiliser des commandes Managed File Transfer pour initier le transfert d'un fichier ou de plusieurs fichiers depuis un noeud Connect:Direct vers un agent MFT.
3. Utiliser des commandes Managed File Transfer pour initier le transfert d'un fichier qui lance un processus Connect:Direct défini par l'utilisateur.
4. Utiliser un processus Connect:Direct pour soumettre une demande de transfert de fichier MFT.

Un pont Connect:Direct peut transférer des fichiers vers ou depuis des noeuds Connect:Direct seulement. Le pont Connect:Direct peut transférer des fichiers vers ou depuis son système de fichiers local uniquement dans le cadre d'un transfert soumis par un processus Connect:Direct.

➤ **z/OS** Vous pouvez utiliser le pont Connect:Direct pour effectuer un transfert vers ou depuis un jeu de données qui se trouve sur un noeud Connect:Direct sur un système z/OS. Le comportement est différent de celui des transferts de jeu de données qui n'impliquent que des agents Managed File Transfer. Pour plus d'informations, voir [z/OS Transfert d'jeux de données depuis et vers des noeuds Connect:Direct](#).

### Plateformes prises en charge

Le pont Connect:Direct est constitué d'un agent de pont MFT Connect:Direct et d'un noeud Connect:Direct. L'agent est pris en charge sous Windows et Linux for x86-64. Le noeud est pris en

charge sur les plateformes qui sont prises en charge pour IBM Sterling Connect:Direct for Windows et IBM Sterling Connect:Direct for UNIX. Pour des instructions sur la création de l'agent de pont Connect:Direct et la configuration d'un noeud Connect:Direct avec lequel l'agent doit communiquer, voir [Configuration du pont Connect:Direct](#).

Le pont Connect:Direct peut transférer des fichiers depuis et vers des noeuds Connect:Direct qui s'exécutent dans le cadre de l'installation de service Connect:Direct for Windows, Connect:Direct for UNIX ou Connect:Direct for z/OS. Pour plus de détails sur les versions de Connect:Direct prises en charge, voir la page Web [System Requirements for IBM MQ](#).

L'agent et le noeud qui composent le pont Connect:Direct doivent se trouver sur le même système ou avoir accès au même système de fichiers, par exemple via un montage NFS partagé. Ce système de fichiers est utilisé pour stocker temporairement les fichiers lors des transferts de fichiers impliquant le pont Connect:Direct, dans un répertoire défini par le paramètre **cdTmpDir**. L'agent de pont Connect:Direct et le noeud de pont Connect:Direct doivent pouvoir traiter ce répertoire en utilisant le même chemin d'accès. Par exemple, si l'agent et le noeud se trouvent sur des systèmes Windows distincts, les systèmes doivent utiliser la même lettre d'unité pour monter le système de fichiers partagé. Les configurations suivantes permettent à l'agent et au noeud d'utiliser le même chemin d'accès :

- L'agent et le noeud se trouvent sur le même système, qui exécute Windows ou Linux for x86-64
- L'agent se trouve sous Linux for x86-64 et le noeud sous AIX
- L'agent se trouve sur un système Windows et le noeud sur un autre système Windows

Les configurations suivantes ne permettent pas à l'agent et au noeud d'utiliser le même chemin d'accès :

- L'agent se trouve sous Linux for x86-64 et le noeud sous Windows
- L'agent se trouve sous Windows et le noeud sous UNIX

Tenez compte de cette restriction lorsque vous planifiez votre installation du pont Connect:Direct.

### Concepts associés

[«Reprise et redémarrage pour les transferts vers et depuis les noeuds Connect:Direct»](#), à la page 335  
Il se peut que Managed File Transfer ne puisse pas se connecter à votre noeud IBM Sterling Connect:Direct lors d'un transfert ; par exemple, si le noeud devient indisponible. Soit Managed File Transfer tente de récupérer le transfert, soit le transfert échoue et un message d'erreur est généré.

[«Soumission d'un processus Connect:Direct défini par l'utilisateur à partir d'une demande de transfert de fichier»](#), à la page 336

Vous pouvez soumettre une demande de transfert pour un transfert qui passe par l'agent de pont Connect:Direct qui appelle un processus Connect:Direct défini par l'utilisateur dans le cadre du transfert de fichiers.

[«Utilisation de processus Connect:Direct pour soumettre des demandes de transfert Managed File Transfer»](#), à la page 340

Vous pouvez soumettre une demande de transfert à l'agent de pont Connect:Direct à partir d'un processus Connect:Direct . Managed File Transfer fournit des commandes qui peuvent être appelées à partir d'une instruction **RUN TASK** dans un processus Connect:Direct .

### Tâches associées

Configuration du pont Connect:Direct

[«Transfert d'un fichier vers un noeud Connect:Direct»](#), à la page 328

Vous pouvez transférer un fichier d'un agent Managed File Transfer vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Indiquez un noeud Connect:Direct comme destination du transfert en spécifiant l'agent de pont Connect:Direct comme agent de destination et en spécifiant le fichier de destination sous la forme *connect\_direct\_node\_name:file\_path*.

[«Transfert d'un fichier à partir d'un noeud Connect:Direct»](#), à la page 329

Vous pouvez transférer un fichier d'un noeud Connect:Direct vers un Managed File Transfer Agent à l'aide du pont Connect:Direct . Vous pouvez spécifier un noeud Connect:Direct comme source du transfert en spécifiant l'agent de pont Connect:Direct comme agent source et en spécifiant la spécification source sous la forme *connect\_direct\_node\_name:file\_path*.

[«Transfert de plusieurs fichiers vers un noeud Connect:Direct», à la page 331](#)

Vous pouvez transférer plusieurs fichiers d'un Managed File Transfer Agent vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Pour utiliser un noeud Connect:Direct comme destination du transfert de plusieurs fichiers, spécifiez l'agent de pont Connect:Direct comme agent de destination et indiquez le répertoire de destination sous la forme `connect_direct_node_name:directory_path`.

[«Transferring multiple files from a Connect:Direct node», à la page 332](#)

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

[«Transfert de plusieurs fichiers vers Connect:Direct à l'aide de caractères génériques», à la page 333](#)

Pour transférer plusieurs fichiers d'un agent Managed File Transfer vers un noeud Connect:Direct , utilisez le pont Connect:Direct . Vous pouvez utiliser des caractères génériques dans la spécification source que vous fournissez à la commande **fteCreateTransfer** . Comme pour tous les transferts Managed File Transfer impliquant des caractères génériques, seule la dernière partie du chemin de fichier peut contenir un caractère générique. Par exemple, `/abc/def*` est un chemin de fichier valide et `/abc*/def` n'est pas valide.

[Traitement des incidents liés au pont Connect:Direct](#)

### Référence associée

[fteCreateCDAgent: créez un agent de pont Connect:Direct](#)

[Restrictions de l'agent de pont Connect:Direct](#)

## Transfert d'un fichier vers un noeud Connect:Direct

Vous pouvez transférer un fichier d'un agent Managed File Transfer vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Indiquez un noeud Connect:Direct comme destination du transfert en spécifiant l'agent de pont Connect:Direct comme agent de destination et en spécifiant le fichier de destination sous la forme `connect_direct_node_name:file_path`.

### Avant de commencer

Avant de transférer un fichier, vous devez configurer le pont Connect:Direct , qui est un composant de Managed File Transfer. Pour plus d'informations, voir [Configuration du pont Connect:Direct](#).

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent de pont Connect:Direct est appelé CD\_BRIDGE. L'agent source est appelé FTE\_AGENT et peut être n'importe quelle version de WMQFTE. Le noeud Connect:Direct de destination est appelé CD\_NODE1. Le fichier à transférer se trouve dans le chemin d'accès au fichier `/home/helen/file.log` sur le système où se trouve FTE\_AGENT. Le fichier est transféré dans le chemin d'accès au fichier `/files/data.log` sur le système sur lequel CD\_NODE1 s'exécute.

### Procédure

1. Utilisez la commande de transfert `fteCreate` avec la valeur du paramètre **-df** (fichier de destination) sous la forme `connect_direct_node_name:file_path` et la valeur du paramètre **-da** (agent de destination) spécifiée comme nom de l'agent de pont Connect:Direct .

**Remarque :** Le noeud Connect:Direct spécifié par `connect_direct_node_name` est le noeud vers lequel le fichier doit être transféré, et non le noeud Connect:Direct qui fait partie du pont Connect:Direct .

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
-df CD_NODE1:/files/data.log /home/helen/file.log
```

Pour plus d'informations, voir [fteCreateTransfer](#): démarrage d'un nouveau transfert de fichier.

2. L'agent source FTE\_AGENT transfère le fichier à l'agent de pont Connect:Direct CD\_BRIDGE. Le fichier est temporairement stocké sur le système où s'exécute l'agent de pont Connect:Direct , à



l'emplacement défini par la propriété d'agent `cdTmpDir`. L'agent de pont `Connect:Direct` transfère le fichier au noeud `Connect:Direct CD_NODE1`.

### Concepts associés

«Le pont `Connect:Direct`», à la page 326

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling `Connect:Direct` existant. Utilisez le pont `Connect:Direct`, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling `Connect:Direct`.

### Tâches associées

«Transfert d'un fichier à partir d'un noeud `Connect:Direct`», à la page 329

Vous pouvez transférer un fichier d'un noeud `Connect:Direct` vers un Managed File Transfer Agent à l'aide du pont `Connect:Direct`. Vous pouvez spécifier un noeud `Connect:Direct` comme source du transfert en spécifiant l'agent de pont `Connect:Direct` comme agent source et en spécifiant la spécification source sous la forme `connect_direct_node_name:file_path`.

### Référence associée

Le fichier `MFT agent.properties`

## Transfert d'un fichier à partir d'un noeud `Connect:Direct`

Vous pouvez transférer un fichier d'un noeud `Connect:Direct` vers un Managed File Transfer Agent à l'aide du pont `Connect:Direct`. Vous pouvez spécifier un noeud `Connect:Direct` comme source du transfert en spécifiant l'agent de pont `Connect:Direct` comme agent source et en spécifiant la spécification source sous la forme `connect_direct_node_name:file_path`.

### Avant de commencer

Avant de transférer un fichier, vous devez configurer le pont `Connect:Direct`, qui est un composant de Managed File Transfer. Voir [Configuration du pont `Connect:Direct`](#).

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent de pont `Connect:Direct` est appelé `CD_BRIDGE`. L'agent de destination est appelé `FTE_AGENT` et peut être n'importe quelle version de Managed File Transfer. Le noeud `Connect:Direct` source est appelé `CD_NODE1`. Le fichier à transférer se trouve dans le chemin d'accès au fichier `/home/brian/in.file` sur le système où se trouve `CD_NODE1`. Le fichier est transféré dans le chemin d'accès au fichier `/files/out.file` sur le système où `FTE_AGENT` est en cours d'exécution.

### Procédure

Utilisez la commande **`fteCreateTransfer`** avec la valeur de la spécification source sous la forme `connect_direct_node_name:file_path` et la valeur du paramètre **`-sa`** spécifiée comme nom de l'agent de pont `Connect:Direct`.

**Remarque :** Le noeud `Connect:Direct` spécifié par `connect_direct_node_name` est le noeud à partir duquel le fichier doit être transféré, et non le noeud `Connect:Direct` qui fait partie du pont `Connect:Direct`.  
Exemple :

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_AGENT
                 -df /files/out.file CD_NODE1:/home/brian/in.file
```

Pour plus d'informations, voir [\*\*`fteCreateTransfer`\*\*: démarrage d'un nouveau transfert de fichier](#).

### Résultats

L'agent de pont `Connect:Direct CD_BRIDGE` demande le fichier au noeud `Connect:Direct CD_NODE1`. Le noeud `Connect:Direct` envoie le fichier au pont `Connect:Direct`. Lors du transfert du fichier à partir du noeud `Connect:Direct`, le pont `Connect:Direct` stocke temporairement le fichier à l'emplacement défini par la propriété d'agent `cdTmpDir`. Lorsque le transfert du fichier depuis le noeud `Connect:Direct` vers le pont `Connect:Direct` est terminé, le pont `Connect:Direct` envoie ensuite le fichier à l'agent cible `FTE_AGENT` et le supprime de l'emplacement temporaire.

## Concepts associés

«Le pont Connect:Direct», à la page 326

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.

## Référence associée

[Le fichier MFT agent.properties](#)

## Transfert d'un ensemble de données vers un noeud Connect:Direct sur z/OS

Vous pouvez transférer un ensemble de données d'un agent Managed File Transfer sous z/OS vers un noeud Connect:Direct sous z/OS à l'aide d'un pont Connect:Direct situé sur un système Windows ou Linux .

## Avant de commencer

Avant de transférer un fichier, vous devez configurer le pont Connect:Direct , qui est un composant de Managed File Transfer. Voir [Configuration du pont Connect:Direct](#).

## Pourquoi et quand exécuter cette tâche

Dans cet exemple, le paramètre **-df** est utilisé pour spécifier la destination du transfert. Le paramètre **-df** peut être utilisé lorsque l'agent source du transfert est une version quelconque de Managed File Transfer. Vous pouvez utiliser le paramètre **-ds** à la place. L'agent source est appelé FTE\_ZOS1 et est un agent Managed File Transfer . L'agent de pont Connect:Direct est appelé CD\_BRIDGE et se trouve sur un système Linux . Le noeud Connect:Direct de destination est appelé CD\_ZOS2. L'agent source et le noeud Connect:Direct de destination se trouvent sur les systèmes z/OS . Le fichier à transférer se trouve dans //FTEUSER . SOURCE . LIB sur le système où se trouve FTE\_ZOS1 . Le fichier est transféré au fichier //CDUSER . DEST . LIB sur le système où se trouve CD\_ZOS2 .

## Procédure

1. Utilisez la commande de transfert `fteCreateavec` la valeur du paramètre **-df** sous la forme `connect_direct_node_name:data_set_name;attributes` et la valeur du paramètre **-da** (agent de destination) spécifiée comme nom de l'agent de pont Connect:Direct .

Le noeud Connect:Direct spécifié par `connect_direct_node_name` est le noeud vers lequel vous souhaitez transférer l'ensemble de données, et non le noeud Connect:Direct qui fait partie du pont Connect:Direct .

Le nom de fichier spécifié par `nom_fichier` doit être absolu et non relatif. Connect:Direct ne préfixe pas le nom de l'ensemble de données avec le nom de l'utilisateur.

```
fteCreateTransfer -sa FTE_ZOS1 -sm QM_ZOS
                  -da CD_BRIDGE -dm QM_BRIDGE
                  -df CD_ZOS2:// 'CDUSER.DEST.LIB;BLKSIZE(8000);LRECL(80)'
                  // 'FTEUSER.SOURCE.LIB'
```

Pour plus d'informations, voir [fteCreateTransfer: démarrage d'un nouveau transfert de fichier](#).

2. L'agent source FTE\_ZOS1 transfère les données du jeu de données à l'agent de pont Connect:Direct CD\_BRIDGE. Les données sont temporairement stockées sous forme de fichier à plat sur le système où s'exécute l'agent de pont Connect:Direct , à l'emplacement défini par la propriété d'agent `cdTmpDir`. L'agent de pont Connect:Direct transfère les données au noeud Connect:Direct CD\_ZOS2. Une fois le transfert terminé, le fichier à plat est supprimé du système sur lequel l'agent de pont Connect:Direct est en cours d'exécution.

## Concepts associés

«Le pont Connect:Direct», à la page 326

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.

### Tâches associées

[Transfert de fichiers depuis et vers des noeuds Connect:Direct](#)

### Référence associée

[Propriétés BPXWDYN que vous ne devez pas utiliser avec MFT](#)

## Transfert de plusieurs fichiers vers un noeud Connect:Direct

Vous pouvez transférer plusieurs fichiers d'un Managed File Transfer Agent vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Pour utiliser un noeud Connect:Direct comme destination du transfert de plusieurs fichiers, spécifiez l'agent de pont Connect:Direct comme agent de destination et indiquez le répertoire de destination sous la forme *connect\_direct\_node\_name:directory\_path*.

### Avant de commencer

Avant de transférer des fichiers, vous devez configurer le pont Connect:Direct , qui est un composant de Managed File Transfer. Voir [Configuration du pont Connect:Direct](#).

### Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent source est appelé FTE\_AGENT. L'agent de pont Connect:Direct est appelé CD\_BRIDGE. Le noeud Connect:Direct de destination est appelé CD\_NODE1. Les fichiers à transférer sont /home/jack/data.log, /logs/log1.txt et /results/latest sur le système où se trouve FTE\_AGENT. Les fichiers sont transférés vers le répertoire /in/files sur le système sur lequel CD\_NODE1 s'exécute.

### Procédure

Utilisez la commande de transfert fteCreate avec la valeur du paramètre **-dd** (répertoire de destination) sous la forme *connect\_direct\_node\_name:directory\_path*. Indiquez la valeur du paramètre **-da** (agent de destination) comme nom de l'agent de pont Connect:Direct .

**Remarque :** Le noeud Connect:Direct spécifié par *connect\_direct\_node\_name* est le noeud vers lequel vous souhaitez que les fichiers soient transférés, et non le noeud Connect:Direct qui fait partie du pont Connect:Direct .

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE
                  -dd CD_NODE1:/in/files /home/jack/data.log
                  /logs/log1.txt /results/latest
```

Pour plus d'informations, voir [fteCreateTransfer: démarrage d'un nouveau transfert de fichier](#).

### Résultats

L'agent source FTE\_AGENT transfère le premier fichier à l'agent de pont Connect:Direct CD\_BRIDGE. L'agent de pont Connect:Direct stocke temporairement le fichier à l'emplacement défini par la propriété de répertoire cdTmp. Lorsque le fichier a été intégralement transféré de l'agent source vers le pont Connect:Direct , l'agent de pont Connect:Direct envoie le fichier au noeud Connect:Direct défini par la propriété d'agent cdNode . Ce noeud envoie le fichier au noeud Connect:Direct de destination CD\_NODE1. L'agent de pont Connect:Direct supprime le fichier de l'emplacement temporaire lorsque le transfert entre les deux noeuds Connect:Direct est terminé. Ce processus est répété pour chaque fichier source spécifié.

### Concepts associés

«Le pont Connect:Direct», à la page 326

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.

## Tâches associées

«Transfert d'un fichier vers un noeud Connect:Direct», à la page 328

Vous pouvez transférer un fichier d'un agent Managed File Transfer vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Indiquez un noeud Connect:Direct comme destination du transfert en spécifiant l'agent de pont Connect:Direct comme agent de destination et en spécifiant le fichier de destination sous la forme `connect_direct_node_name:file_path`.

«Transfert de plusieurs fichiers vers Connect:Direct à l'aide de caractères génériques», à la page 333

Pour transférer plusieurs fichiers d'un agent Managed File Transfer vers un noeud Connect:Direct , utilisez le pont Connect:Direct . Vous pouvez utiliser des caractères génériques dans la spécification source que vous fournissez à la commande **fteCreateTransfer** . Comme pour tous les transferts Managed File Transfer impliquant des caractères génériques, seule la dernière partie du chemin de fichier peut contenir un caractère générique. Par exemple, `/abc/def*` est un chemin de fichier valide et `/abc*/def` n'est pas valide.

«Transfert d'un fichier à partir d'un noeud Connect:Direct», à la page 329

Vous pouvez transférer un fichier d'un noeud Connect:Direct vers un Managed File Transfer Agent à l'aide du pont Connect:Direct . Vous pouvez spécifier un noeud Connect:Direct comme source du transfert en spécifiant l'agent de pont Connect:Direct comme agent source et en spécifiant la spécification source sous la forme `connect_direct_node_name:file_path`.

«Transferring multiple files from a Connect:Direct node», à la page 332

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

## Référence associée

Le fichier MFT `agent.properties`

## Transferring multiple files from a Connect:Direct node

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form `connect_direct_node_name:file_path`.

## Before you begin

Before transferring a file, you must configure the Connect:Direct bridge, which is a component of Managed File Transfer. See [Configuring the Connect:Direct bridge](#).

## About this task

In this example, the Connect:Direct bridge agent is called CD\_BRIDGE. The destination agent is called FTE\_Z, and is running on a z/OS system. The source Connect:Direct node is called CD\_NODE1. The files to be transferred are located at the file paths `/in/file1`, `/in/file2`, and `/in/file3` on the system where CD\_NODE1 is located. The files are transferred to the partitioned data set `//OBJECT.LIB` on the system where FTE\_Z is running.

## Procedure

Use the `fteCreateTransfer` command with the values for the source specifications in the form `connect_direct_node_name:file_path` and the value of the `-sa` parameter specified as the name of the Connect:Direct bridge agent.

**Note:** The Connect:Direct node specified by *connect\_direct\_node\_name* is the node that you want the files to be transferred from, not the Connect:Direct node that operates as part of the Connect:Direct bridge.

```
fteCreateTransfer -sa CD_BRIDGE -da FTE_Z  
-dp //'OBJECT.LIB' CD_NODE1:/in/file1  
CD_NODE1:/in/file2 CD_NODE1:/in/file3
```

For more information, see [fteCreateTransfer: start a new file transfer](#).

## Results

The Connect:Direct bridge agent CD\_BRIDGE requests the first file from the Connect:Direct node CD\_NODE1. The Connect:Direct node sends the file to the Connect:Direct bridge. While the file is being transferred from the Connect:Direct node, the Connect:Direct bridge stores the file temporarily in the location defined by the cdTmpDir agent property. When the file has finished transferring from the Connect:Direct node to the Connect:Direct bridge, the Connect:Direct bridge sends the file to the destination agent FTE\_Z and then deletes the file from the temporary location. This process is repeated for each specified source file.

### Related concepts

[“Le pont Connect:Direct” on page 326](#)

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.

### Related reference

[The MFT agent.properties file](#)

## Transfert de plusieurs fichiers vers Connect:Direct à l'aide de caractères génériques

Pour transférer plusieurs fichiers d'un agent Managed File Transfer vers un noeud Connect:Direct, utilisez le pont Connect:Direct. Vous pouvez utiliser des caractères génériques dans la spécification source que vous fournissez à la commande **fteCreateTransfer**. Comme pour tous les transferts Managed File Transfer impliquant des caractères génériques, seule la dernière partie du chemin de fichier peut contenir un caractère générique. Par exemple, `/abc/def*` est un chemin de fichier valide et `/abc*/def` n'est pas valide.

## Avant de commencer

Avant de transférer un fichier, vous devez configurer le pont Connect:Direct, qui est un composant de Managed File Transfer. Pour plus d'informations, voir [Configuration du pont Connect:Direct](#).

## Pourquoi et quand exécuter cette tâche

Dans cet exemple, l'agent source est appelé FTE\_AGENT et l'agent de pont Connect:Direct est appelé CD\_BRIDGE. Le noeud Connect:Direct de destination est appelé CD\_NODE1. Les fichiers à transférer se trouvent dans le répertoire `/reports` sur le système où se trouve FTE\_AGENT. Seuls les fichiers dont le nom commence par `report`, suivi de deux caractères et du suffixe `.log`, sont transférés. Par exemple, le fichier `/reports/report01.log` est transféré, mais le fichier `/reports/report1.log` n'est pas transféré. Les fichiers sont transférés vers le répertoire `/home/fred` sur le système sur lequel CD\_NODE1 s'exécute.

## Procédure

1. Utilisez la commande de transfert `fteCreate` avec la valeur du paramètre **-dd** (répertoire de destination) sous la forme `connect_direct_node_name:directory_path`. Pour le paramètre **-da** (agent de destination), spécifiez l'agent de pont Connect:Direct.

**Remarque :** Le noeud Connect:Direct spécifié par *connect\_direct\_node\_name* est le noeud vers lequel vous souhaitez que les fichiers soient transférés, et non le noeud Connect:Direct qui fait partie du pont Connect:Direct .

```
fteCreateTransfer -sa FTE_AGENT -da CD_BRIDGE  
-dd CD_NODE1:/home/fred "/reports/report??.log"
```

Pour plus d'informations, voir **fteCreateTransfer**: démarrage d'un nouveau transfert de fichier.

2. L'agent source FTE\_AGENT transfère le premier fichier qui correspond au modèle */reports/report??.log* à l'agent de pont Connect:Direct CD\_BRIDGE. L'agent de pont Connect:Direct stocke temporairement le fichier à l'emplacement défini par la propriété de répertoire *cdTmp*. Lorsque le fichier a été intégralement transféré de l'agent source vers le pont Connect:Direct , l'agent de pont Connect:Direct envoie le fichier au noeud Connect:Direct défini par la propriété d'agent *cdNode* . Ce noeud envoie le fichier au noeud Connect:Direct de destination CD\_NODE1. L'agent de pont Connect:Direct supprime le fichier de l'emplacement temporaire lorsque le transfert entre les deux noeuds Connect:Direct est terminé.Ce processus est répété pour chaque fichier source qui correspond au modèle de caractère générique */reports/report??.log*.

**Remarque :** La liste des fichiers correspondant au modèle */reports/report??.log* varie en fonction du système d'exploitation du système sur lequel se trouve l'agent source FTE\_AGENT.

- Si l'agent source se trouve sur un système avec un système d'exploitation Windows , la correspondance de modèle n'est pas sensible à la casse. Le masque correspond à tous les fichiers du répertoire */reports* avec un nom de fichier au format *report* suivi de deux caractères et d'un suffixe *.log*, quelle que soit la casse des lettres. Par exemple, *Report99.Log* est une correspondance.
- Si l'agent source se trouve sur un système Linux ou UNIX , la correspondance de modèle est sensible à la casse. Le modèle correspond uniquement aux fichiers du répertoire */reports* avec un nom de fichier au format *report* suivi de deux caractères et d'un suffixe *.log*. Par exemple, *reportAB.log* est une correspondance, mais *reportAB.LOG* et *Report99.Log* ne sont pas des correspondances.

### Concepts associés

«Le pont Connect:Direct», à la page 326

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.

### Tâches associées

Utilisation de caractères génériques avec MFT

«Transfert d'un fichier vers un noeud Connect:Direct», à la page 328

Vous pouvez transférer un fichier d'un agent Managed File Transfer vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Indiquez un noeud Connect:Direct comme destination du transfert en spécifiant l'agent de pont Connect:Direct comme agent de destination et en spécifiant le fichier de destination sous la forme *connect\_direct\_node\_name:file\_path*.

«Transfert de plusieurs fichiers vers un noeud Connect:Direct», à la page 331

Vous pouvez transférer plusieurs fichiers d'un Managed File Transfer Agent vers un noeud Connect:Direct à l'aide du pont Connect:Direct . Pour utiliser un noeud Connect:Direct comme destination du transfert de plusieurs fichiers, spécifiez l'agent de pont Connect:Direct comme agent de destination et indiquez le répertoire de destination sous la forme *connect\_direct\_node\_name:directory\_path*.

«Transferring multiple files from a Connect:Direct node», à la page 332

You can transfer multiple files from a Connect:Direct node to a Managed File Transfer Agent by using the Connect:Direct bridge. You can specify a Connect:Direct node as the source of the multiple file transfer by specifying the Connect:Direct bridge agent as the source agent and specifying one or more source specifications in the form *connect\_direct\_node\_name:file\_path*.

### Référence associée

Le fichier MFT agent.properties

## Reprise et redémarrage pour les transferts vers et depuis les noeuds Connect:Direct

Il se peut que Managed File Transfer ne puisse pas se connecter à votre noeud IBM Sterling Connect:Direct lors d'un transfert ; par exemple, si le noeud devient indisponible. Soit Managed File Transfer tente de récupérer le transfert, soit le transfert échoue et un message d'erreur est généré.

### Si le noeud Connect:Direct devient indisponible

Si le noeud Connect:Direct devient indisponible ; par exemple, en raison d'une panne de réseau ou d'alimentation, Managed File Transfer récupère un transfert de fichiers de l'une des manières suivantes:

- Si Managed File Transfer n'a pas réussi à se connecter au noeud Connect:Direct dans le cadre de cette demande de transfert, le transfert est relancé pendant une période déterminée par les valeurs de **cdMaxConnectionRetries** et **recoverableTransferRetryInterval properties**. Ces propriétés sont spécifiées dans le fichier `agent.properties` pour l'agent de pont Connect:Direct. Le transfert échoue et un message d'erreur est généré lorsque le nombre de tentatives ayant échoué atteint la valeur de **cdMaxConnectionRetries property**. Par défaut, le transfert est tenté indéfiniment, avec 60 secondes entre les tentatives.
- Si Managed File Transfer a précédemment réussi à se connecter au noeud Connect:Direct dans le cadre de cette demande de transfert, le transfert est retenté pendant une période déterminée par les valeurs des propriétés **cdMaxPartialWorkConnectionRetries** et **recoverableTransferRetryInterval**. Le transfert échoue et un message d'erreur est généré lorsque le nombre de tentatives ayant échoué atteint la valeur de la propriété **cdMaxPartialWorkConnectionRetries**. Par défaut, le transfert est tenté indéfiniment, avec 60 secondes entre les tentatives.
- Pour certains types d'échec de noeud Connect:Direct, par exemple le noeud en cours d'arrêt forcé, les processus Connect:Direct passent à l'état `HEld Due to Error (HE)` lors de la reprise du noeud. Une fois le noeud restauré, Managed File Transfer reprend automatiquement tous les processus Connect:Direct liés au transfert de fichiers et dont le statut est HE.
- Si le transfert échoue, tous les fichiers temporaires associés au transfert sont supprimés du système qui héberge le pont Connect:Direct. L'emplacement de ces fichiers temporaires est défini par la propriété **cdTmpDir**.
- Si le transfert est de Managed File Transfer vers Connect:Direct et qu'une disposition source de suppression est spécifiée, les fichiers source ne sont pas supprimés si le transfert échoue.

### Si les données d'identification de l'utilisateur du noeud Connect:Direct ne sont pas valides

Si Managed File Transfer ne parvient pas à se connecter au noeud Connect:Direct car les données d'identification de l'utilisateur sont rejetées par le noeud, le transfert échoue et un message d'erreur est généré. Dans ce cas, vérifiez que vous avez fourni les données d'identification utilisateur correctes pour le noeud Connect:Direct. Pour plus d'informations, voir [Mappage des données d'identification pour Connect:Direct](#).

### Si l'agent de pont Connect:Direct devient indisponible

Si l'agent de pont Connect:Direct devient indisponible, les transferts de fichiers en cours sont effectués de la même manière que les transferts Managed File Transfer standard. Pour plus d'informations, voir [«MFT Reprise et redémarrage»](#), à la page 343.

#### Concepts associés

«Le pont Connect:Direct», à la page 326

Vous pouvez transférer des fichiers depuis et vers un réseau IBM Sterling Connect:Direct existant. Utilisez le pont Connect:Direct, qui est un composant de Managed File Transfer, pour transférer des fichiers entre MFT et IBM Sterling Connect:Direct.

«MFT Reprise et redémarrage», à la page 343

Si votre agent ou votre gestionnaire de files d'attente n'est pas disponible pour une raison quelconque, par exemple en raison d'une panne d'alimentation ou de réseau, Managed File Transfer effectue la récupération suivante dans les scénarios suivants:

#### **Tâches associées**

[Configuration du pont Connect:Direct](#)

#### **Référence associée**

[Le fichier MFT agent.properties](#)

## **Soumission d'un processus Connect:Direct défini par l'utilisateur à partir d'une demande de transfert de fichier**

Vous pouvez soumettre une demande de transfert pour un transfert qui passe par l'agent de pont Connect:Direct qui appelle un processus Connect:Direct défini par l'utilisateur dans le cadre du transfert de fichiers.

Par défaut, lorsque vous soumettez une demande de transfert de fichier pour un transfert qui passe par le pont Connect:Direct, l'agent de pont Connect:Direct génère le processus Connect:Direct qui est utilisé pour transférer le fichier vers ou depuis le noeud Connect:Direct distant.

Toutefois, vous pouvez configurer l'agent de pont Connect:Direct pour qu'il appelle à la place un processus Connect:Direct défini par l'utilisateur à l'aide du fichier `ConnectDirectProcessDefinition.xml`.

### **Le fichier `ConnectDirectProcessDefinition.xml`**

La commande **fteCreateCDAgent** crée le fichier `ConnectDirectProcessDefinitions.xml` dans le répertoire de configuration de l'agent `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Avant de pouvoir appeler des processus Connect:Direct définis par l'utilisateur à partir de l'agent de pont Connect:Direct, vous devez configurer des définitions de processus en éditant ce fichier.

Le fichier définit un ou plusieurs ensembles de processus qui incluent l'emplacement d'un ou de plusieurs processus Connect:Direct appelés dans le cadre d'un transfert. Chaque ensemble de processus inclut un certain nombre de conditions. Si le transfert remplit toutes les conditions de l'ensemble de processus, l'ensemble de processus est utilisé pour spécifier les processus Connect:Direct qui sont appelés par le transfert. Pour plus d'informations, voir [«Spécification du processus Connect:Direct à démarrer à l'aide du fichier `ConnectDirectProcessDefinition.xml`»](#), à la page 337.

### **Variables symboliques intrinsèques**

Vous pouvez utiliser les variables symboliques intrinsèques définies par Managed File Transfer pour remplacer des valeurs dans des processus Connect:Direct définis par l'utilisateur. Pour respecter la convention de dénomination Connect:Direct, toutes les variables symboliques intrinsèques utilisées par Managed File Transfer ont le format `%FTE` suivi de cinq caractères alphanumériques en majuscules.

Lors de la création d'un processus pour transférer des fichiers depuis un noeud Connect:Direct sur le système de pont Connect:Direct, vous devez utiliser la variable intrinsèque `%FTETFILE` comme valeur pour TO FILE dans le processus Connect:Direct. Lors de la création d'un processus pour transférer des fichiers sur un noeud Connect:Direct depuis le système de pont Connect:Direct, vous devez utiliser la variable intrinsèque `%FTEFFILE` comme valeur de FROM FILE dans le processus Connect:Direct. Ces variables contiennent les chemins d'accès aux fichiers temporaires que l'agent de pont Connect:Direct utilise pour les transferts vers et depuis le réseau Managed File Transfer.

Pour plus d'informations sur les variables symboliques intrinsèques, voir la documentation du produit Connect:Direct.



## Exemples de processus Connect:Direct

Managed File Transfer fournit des exemples de processus Connect:Direct . Ces exemples se trouvent dans le répertoire suivant: `MQ_INSTALLATION_PATH/mqft/samples/ConnectDirectProcessTemplates`.

### Tâches associées

«[Spécification du processus Connect:Direct à démarrer à l'aide du fichier ConnectDirectProcessDefinition.xml](#)», à la page 337

Indiquez le processus Connect:Direct à démarrer dans le cadre d'un transfert Managed File Transfer . Managed File Transfer fournit un fichier XML que vous pouvez éditer pour spécifier des définitions de processus.

«[Utilisation de variables symboliques intrinsèques dans les processus Connect:Direct appelés par Managed File Transfer](#)», à la page 338

Vous pouvez appeler un processus Connect:Direct défini par l'utilisateur à partir d'un transfert Managed File Transfer et transmettre des informations du transfert au processus Connect:Direct en utilisant des variables symboliques intrinsèques dans la définition de processus.

### Référence associée

[Format de fichier de définition de processus Connect:Direct](#)

[Variables de substitution à utiliser avec les processus Connect:Direct définis par l'utilisateur](#)

### ***Spécification du processus Connect:Direct à démarrer à l'aide du fichier ConnectDirectProcessDefinition.xml***

Indiquez le processus Connect:Direct à démarrer dans le cadre d'un transfert Managed File Transfer . Managed File Transfer fournit un fichier XML que vous pouvez éditer pour spécifier des définitions de processus.

## Pourquoi et quand exécuter cette tâche

La commande **fteCreateCDAgent** crée le fichier `ConnectDirectProcessDefinitions.xml` dans le répertoire de configuration de l'agent `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name`. Avant de pouvoir appeler des processus Connect:Direct définis par l'utilisateur à partir de l'agent de pont Connect:Direct , vous devez configurer des définitions de processus en éditant ce fichier.

Pour chaque processus à appeler dans le cadre d'un transfert via le pont Connect:Direct , procédez comme suit:

### Procédure

1. Définissez le processus Connect:Direct que l'agent de pont Connect:Direct doit appeler dans le cadre du transfert et sauvegardez le modèle de processus dans un fichier.
2. Ouvrez le fichier `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name/ConnectDirectProcessDefinitions.xml` dans n'importe quel éditeur de texte.
3. Créez un élément `<processSet>` .
4. Dans l'élément `<processSet>` , créez un élément `<condition>` .
5. Dans l'élément `<condition>` , créez un ou plusieurs éléments qui définissent une condition à laquelle la demande de transfert doit correspondre pour appeler le processus Connect:Direct que vous avez défini à l'étape 1. Ces éléments peuvent être des éléments `<match>` ou des éléments `<defined>` .
  - Utilisez un élément `<match>` pour indiquer que la valeur d'une variable doit correspondre à un modèle. Créez l'élément `<match>` avec les attributs suivants:

- `variable` -nom de la variable dont la valeur est comparée. La variable est un symbole intrinsèque. Pour plus d'informations, voir [Variables de substitution à utiliser avec les processus Connect:Direct définis par l'utilisateur](#).
- `value` -modèle à comparer à la valeur de la variable spécifiée.
- Facultatif: `pattern` -type de canevas utilisé par la valeur de l'attribut `value` . Ce type de canevas peut être wildcard ou regex. Cet attribut est facultatif et la valeur par défaut est wildcard.
- Utilisez un élément `<defined>` pour indiquer qu'une variable doit avoir une valeur définie. Créez l'élément `<defined>` avec l'attribut suivant:
  - `variable` -nom de la variable pour laquelle une valeur doit être définie. La variable est un symbole intrinsèque. Pour plus d'informations, voir [Variables de substitution à utiliser avec les processus Connect:Direct définis par l'utilisateur](#).

Les conditions spécifiées dans l'élément `<condition>` sont combinées avec un ET logique. Toutes les conditions doivent être remplies pour que l'agent de pont Connect:Direct puisse appeler le processus spécifié par cet élément `<processSet>` . Si vous ne spécifiez pas d'élément `<condition>` , l'ensemble de processus correspond à tous les transferts.

6. Dans l'élément `<processSet>` , créez un élément `<process>` .

7. Dans l'élément `<process>` , créez un élément `<transfer>` .

L'élément `transfer` spécifie le processus Connect:Direct que l'agent de pont Connect:Direct appelle dans le cadre du transfert. Créez l'élément `<transfer>` avec l'attribut suivant:

- `process` - emplacement du processus Connect:Direct que vous avez défini à l'étape 1. L'emplacement de ce fichier est spécifié avec un chemin d'accès absolu ou relatif au répertoire `MQ_DATA_PATH/mqft/config/coordination_queue_manager/agents/cd_bridge_agent_name` .

## Résultats

Lors de la recherche d'une correspondance de condition, l'agent de pont Connect:Direct effectue une recherche depuis le début du fichier jusqu'à la fin du fichier. La première correspondance trouvée est celle qui est utilisée.

### Tâches associées

[Configuration du pont Connect:Direct](#)

### Référence associée

[Format de fichier de définition de processus Connect:Direct](#)

[fteCreateCDAgent: créez un agent de pont Connect:Direct](#)

## ***Utilisation de variables symboliques intrinsèques dans les processus Connect:Direct appelés par Managed File Transfer***

Vous pouvez appeler un processus Connect:Direct défini par l'utilisateur à partir d'un transfert Managed File Transfer et transmettre des informations du transfert au processus Connect:Direct en utilisant des variables symboliques intrinsèques dans la définition de processus.

### **Pourquoi et quand exécuter cette tâche**

Cet exemple utilise des variables symboliques intrinsèques pour transmettre des informations d'un transfert Managed File Transfer vers un processus Connect:Direct défini par l'utilisateur. Pour plus d'informations sur les variables symboliques intrinsèques utilisées par Managed File Transfer, voir [Variables de substitution à utiliser avec les processus Connect:Direct définis par l'utilisateur](#).

Dans cet exemple, le fichier est transféré d'un Managed File Transfer Agent à un noeud de pont Connect:Direct . La première partie du transfert est effectuée par Managed File Transfer. La deuxième partie du transfert est effectuée par un processus Connect:Direct défini par l'utilisateur.

## Procédure

1. Créez un processus Connect:Direct qui utilise des variables symboliques intrinsèques.

```
%FTEPNAME PROCESS
  SNODE=%FTESNODE
  PNODEID=(%FTEPUSER,%FTEPPASS)
  SNODEID=(%FTESUSER,%FTESPASS)

COPY001 COPY
  FROM (
    FILE=%FTEFFILE
    DISP=%FTEFDISP
  )
  TO (
    FILE=%FTETFILE
    DISP=%FTETDISP
  )
PEND
```

2. Sauvegardez ce processus dans un fichier texte à l'emplacement suivant: *MQ\_DATA\_PATH/mqft/config/coordination\_queue\_manager/agents/cd\_bridge\_agent/Example.cdp*
3. Modifiez le fichier *ConnectDirectProcessDefinition.xml* pour inclure une règle qui appelle le processus Connect:Direct que vous avez créé à l'étape 1.

```
<?xml version="1.0" encoding="UTF-8"?>
<tns:cdprocess xmlns:tns="http://wmqfte.ibm.com/ConnectDirectProcessDefinitions"
  xmlns:xsi="https://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://wmqfte.ibm.com/
  ConnectDirectProcessDefinitions ConnectDirectProcessDefinitions.xsd">

  <tns:processSet>
    <tns:condition>
      <tns:match variable="%FTESNODE" value="TOBERMORY" pattern="wildcard" />
    </tns:condition>
    <tns:process>
      <tns:transfer process="Example.cdp" />
    </tns:process>
  </tns:processSet>

</tns:cdprocess>
```

Dans cet exemple, si une demande de transfert est soumise à l'agent de pont Connect:Direct dont le noeud Connect:Direct source ou de destination est TOBERMORY, le processus *Example.cdp* Connect:Direct est appelé.

4. Soumettez une demande de transfert de fichier qui répond aux conditions que vous avez définies dans le fichier *ConnectDirectProcessDefinition.xml* à l'étape 3.

Exemple :

```
fteCreateTransfer -sa ORINOCO -da CD_BRIDGE
                  -sm QM_WIMBLEDON -dm QM_COMMON
                  -de overwrite -df TOBERMORY:/home/bulgaria/destination.txt
                  -sd leave c:\bungo\source.txt
```

Dans cet exemple, le noeud Connect:Direct de destination est TOBERMORY. Ce noeud est le noeud secondaire du transfert et la valeur de %FTESNODE est définie sur TOBERMORY. Cette commande correspond à la condition définie dans le fichier *ConnectDirectProcessDefinition.xml*.

5. Managed File Transfer transfère le fichier source à un emplacement temporaire sur le même système que l'agent de pont Connect:Direct.
6. L'agent de pont Connect:Direct définit les valeurs des variables symboliques intrinsèques à partir des informations de la demande de transfert et des informations de configuration.

Les variables symboliques intrinsèques sont définies sur les valeurs suivantes:

- %FTEPNAME=*process\_name* - Cette valeur est un nom de processus de 8 caractères généré par l'agent de pont Connect:Direct.

- %FTESNODE=TOBERMORY-Cette valeur est définie à partir du paramètre **-df** de la commande **fteCreateTransfer** .
  - %FTEPUSER, =*primary\_node\_user* -Ces informations sont extraites du fichier ConnectDirectCredentials.xml .
  - %FTEPPASS=*primary\_node\_user\_password* -Ces informations proviennent du fichier ConnectDirectCredentials.xml .
  - %FTESUSER, =*secondary\_node\_user* -Ces informations sont extraites du fichier ConnectDirectCredentials.xml .
  - %FTESPASS=*secondary\_node\_user\_password* -Ces informations sont extraites du fichier ConnectDirectCredentials.xml .
  - %FTEFFILE =*emplacement\_temporaire* -Cette valeur correspond à l'emplacement temporaire du fichier sur le même système que l'agent de pont Connect:Direct .
  - %FTEFDISP=leave-Cette valeur est définie à partir du paramètre **-sd** de la commande **fteCreateTransfer** .
  - %FTETFILE=/home/bulgaria/destination.txt -Cette valeur est définie à partir du paramètre **-df** de la commande **fteCreateTransfer** .
  - %FTETDISP=overwrite-Cette valeur est définie à partir du paramètre **-de** de la commande **fteCreateTransfer** .
7. Le processus Connect:Direct est démarré sur le noeud de pont Connect:Direct . Connect:Direct transfère le fichier de l'emplacement temporaire sur le système de pont Connect:Direct vers la destination /home/bulgaria/destination.txt sur le système où le noeud Connect:Direct TOBERMORY est en cours d'exécution.

### Concepts associés

«Soumission d'un processus Connect:Direct défini par l'utilisateur à partir d'une demande de transfert de fichier», à la page 336

Vous pouvez soumettre une demande de transfert pour un transfert qui passe par l'agent de pont Connect:Direct qui appelle un processus Connect:Direct défini par l'utilisateur dans le cadre du transfert de fichiers.

### Référence associée

Variables de substitution à utiliser avec les processus Connect:Direct définis par l'utilisateur

## Utilisation de processus Connect:Direct pour soumettre des demandes de transfert Managed File Transfer

Vous pouvez soumettre une demande de transfert à l'agent de pont Connect:Direct à partir d'un processus Connect:Direct . Managed File Transfer fournit des commandes qui peuvent être appelées à partir d'une instruction **RUN TASK** dans un processus Connect:Direct .

Managed File Transfer fournit les commandes suivantes à utiliser avec les processus Connect:Direct :

### ftetag

Spécifiez cette commande dans une étape qui précède la commande **ftebxfer** ou **ftecxfer** pour créer les informations d'audit requises pour le transfert. Cette commande prend la spécification source du transfert comme paramètre. Pour plus d'informations sur le format de la spécification source, voir **fteCreateTransfer**: démarrage d'un nouveau transfert de fichier.

### ftebxfer

Indiquez cette commande pour créer une demande de transfert de fichier lorsque le gestionnaire de files d'attente auquel la demande de transfert est soumise se trouve sur le même système que le noeud Connect:Direct qui soumet la commande. Cette commande utilise les mêmes paramètres que la commande **fteCreateTransfer** . Pour plus d'informations sur ces paramètres, voir **fteCreateTransfer**: démarrage d'un nouveau transfert de fichier. Cette commande comporte également un paramètre supplémentaire:

#### **-qmgrname**

Obligatoire. Nom du gestionnaire de files d'attente auquel la commande doit être soumise.

## **ftecxfer**

Indiquez cette commande pour créer une demande de transfert de fichier lorsque le gestionnaire de files d'attente auquel la demande de transfert est soumise se trouve sur un système différent du noeud Connect:Direct qui soumet la commande. Cette commande utilise les mêmes paramètres que la commande **fteCreateTransfer** . Pour plus d'informations sur les paramètres, voir **fteCreateTransfer: démarrer un nouveau transfert de fichier**. Cette commande comporte également trois paramètres supplémentaires:

### **-qmgrname**

Obligatoire. Nom du gestionnaire de files d'attente auquel la commande doit être soumise.

### **-connname**

Obligatoire. Hôte et port du gestionnaire de files d'attente sur lequel la commande doit être soumise, spécifié au format IBM MQ CONNAME. Par exemple, host.example.com(1337).

### **-channelname**

Facultatif. Nom du canal à utiliser pour la connexion au gestionnaire de files d'attente auquel la commande doit être soumise. S'il n'est pas spécifié, la valeur par défaut est SYSTEM.DEF.SVRCONN est utilisé.

## **Tâches associées**

«Création et soumission d'un processus Connect:Direct qui appelle Managed File Transfer à l'aide de Connect:Direct Requester», à la page 341

Le demandeur Connect:Direct est une interface graphique que vous pouvez utiliser pour créer et soumettre un processus Connect:Direct qui appelle Managed File Transfer.

## **Référence associée**

Exemple: un fichier de processus Connect:Direct qui appelle des commandes MFT

## ***Création et soumission d'un processus Connect:Direct qui appelle Managed File Transfer à l'aide de Connect:Direct Requester***

Le demandeur Connect:Direct est une interface graphique que vous pouvez utiliser pour créer et soumettre un processus Connect:Direct qui appelle Managed File Transfer.

## **Pourquoi et quand exécuter cette tâche**

Cette tâche explique comment créer un processus Connect:Direct qui appelle la commande Managed File Transfer **ftecxfer** ou la commande **ftebxfer** . Utilisez la commande **ftecxfer** lorsque le gestionnaire de files d'attente auquel la demande de transfert est soumise se trouve sur un système différent du noeud Connect:Direct qui soumet la commande. Utilisez la commande **ftebxfer** lorsque le gestionnaire de files d'attente auquel la demande de transfert est soumise se trouve sur le même système que le noeud Connect:Direct qui soumet la commande. La commande **ftecxfer** établit une connexion client au gestionnaire de files d'attente de l'agent source du transfert. Avant d'appeler la commande **ftecxfer** , vous devez appeler la commande **ftetag** et lui transmettre les informations de spécification source. Cela permet au processus d'être consigné et audité de la même manière que les transferts lancés à partir de Managed File Transfer.

## **Procédure**

1. Démarrez le demandeur Connect:Direct .
2. Dans l'onglet **Noeuds** du panneau, sélectionnez le noeud Connect:Direct utilisé comme noeud principal du processus.
3. Sélectionnez **Fichier > Nouveau > processus**. La fenêtre **Propriétés de processus** s'ouvre.
4. Dans la zone **Nom:** , entrez le nom du processus.
5. Sélectionnez le noeud secondaire dans la liste **Snode > Nom:** .
6. Sélectionnez le système d'exploitation du noeud secondaire dans la liste **Snode > Système d'exploitation:** .
7. Facultatif : Renseignez les informations supplémentaires de cette fenêtre dont vous avez besoin.

8. Cliquez sur **OK**. La fenêtre **Propriétés de processus** se ferme.
9. Créez une instruction qui exécute la commande Managed File Transfer **ftetag** .
  - a) Cliquez avec le bouton droit de la souris dans la fenêtre **Processus** de l'instruction **Fin** .
  - b) Sélectionnez **Insérer > Exécuter la tâche**. La fenêtre **Exécuter l'instruction de tâche** s'ouvre.
  - c) Dans la zone **Libellé:** , entrez Tag.
  - d) Dans la zone **Paramètres ou commandes facultatifs** , entrez `pgm(MQ_INSTALLATION_PATH/bin/ftetag) args(source_specification)`. Pour plus d'informations sur le format de *source\_specification*, voir [fteCreateTransfer: démarrage d'un nouveau transfert de fichier](#).
  - e) Cliquez sur **OK**. La fenêtre **Exécuter l'instruction de tâche** se ferme.
10. Créez une instruction qui exécute la commande Managed File Transfer **ftecxfer** ou **ftebxfer** .
  - a) Cliquez avec le bouton droit de la souris dans la fenêtre **Processus** de l'instruction **Fin** .
  - b) Sélectionnez **Insérer > Exécuter la tâche**. La fenêtre **Exécuter l'instruction de tâche** s'ouvre.
  - c) Dans la zone **Libellé:** , entrez Transfer.
  - d) Dans la zone **Paramètres ou commandes facultatifs** , entrez `pgm(MQ_INSTALLATION_PATH/bin/ftecxfer) args(parameters)` ou `pgm(MQ_INSTALLATION_PATH/bin/ftebxfer) args(parameters)` en fonction de la commande que vous choisissez. Les paramètres utilisés par les commandes **ftecxfer** et **ftebxfer** sont les mêmes que ceux utilisés par la commande **fteCreateTransfer** , ainsi que certains paramètres supplémentaires spécifiques à **ftecxfer** et **ftebxfer**. Pour plus d'informations, voir [fteCreateTransfer: démarrez un nouveau transfert de fichier et «Utilisation de processus Connect:Direct pour soumettre des demandes de transfert Managed File Transfer»](#), à la page 340.
  - e) Cliquez sur **OK**. La fenêtre **Exécuter l'instruction de tâche** se ferme.
11. Facultatif : Créez les instructions supplémentaires dont vous avez besoin.
12. Soumettez le processus.
  - a) Cliquez avec le bouton droit de la souris dans la fenêtre **Processus** .
  - b) Sélectionnez **Soumettre**. La fenêtre **Connect:Direct Attach** s'ouvre.
  - c) Entrez le nom d'utilisateur et le mot de passe à utiliser pour exécuter le processus.
  - d) Cliquez sur **OK**.

### Concepts associés

[«Utilisation de processus Connect:Direct pour soumettre des demandes de transfert Managed File Transfer»](#), à la page 340

Vous pouvez soumettre une demande de transfert à l'agent de pont Connect:Direct à partir d'un processus Connect:Direct . Managed File Transfer fournit des commandes qui peuvent être appelées à partir d'une instruction **RUN TASK** dans un processus Connect:Direct .

## Utilisation de MFT à partir de IBM Integration Bus

Vous pouvez utiliser Managed File Transfer depuis IBM Integration Bus à l'aide des noeuds FTEOutput et FTEInput.

- Utilisez le noeud FTEInput pour transférer un fichier sur le réseau à l'aide de Managed File Transfer, puis traiter ce fichier dans le cadre d'un flux Integration Bus.
- Utilisez le noeud FTEOutput pour transférer un fichier généré par un flux Integration Bus à un autre emplacement sur le réseau.

Les agents qui transfèrent des fichiers vers ou depuis l'agent courtier peuvent se trouver à n'importe quel niveau de Managed File Transfer.

Pour plus d'informations, voir la [documentation du produit IBM Integration Bus](#).

## MFT Reprise et redémarrage

Si votre agent ou votre gestionnaire de files d'attente n'est pas disponible pour une raison quelconque, par exemple en raison d'une panne d'alimentation ou de réseau, Managed File Transfer effectue la récupération suivante dans les scénarios suivants:

- Généralement, en cas de problème lors du transfert d'un fichier, Managed File Transfer récupère et redémarre ce transfert de fichier une fois le problème résolu.
- Si un fichier qui était en cours de transfert est supprimé ou modifié alors que l'agent ou le gestionnaire de files d'attente n'est pas disponible, le transfert échoue et vous obtenez un message dans le journal de transfert qui fournit des détails sur l'échec.
- Si un processus d'agent échoue lors d'un transfert de fichier, le transfert se poursuit lorsque vous redémarrez l'agent.
- Si un agent perd la connexion à son gestionnaire de files d'attente d'agent, l'agent attend en tentant de se reconnecter au gestionnaire de files d'attente. Lorsque l'agent se reconnecte à son gestionnaire de files d'attente, le transfert en cours se poursuit.
- Si l'agent est arrêté pour une raison quelconque, tous les moniteurs de ressources associés à une interrogation d'arrêt d'agent. Lorsque l'agent est restauré, les moniteurs sont également redémarrés et l'interrogation des ressources reprend.
- Pour un transfert de fichier avec une disposition source de `delete`, si une récupération se produit après l'envoi de toutes les données d'un agent source à un agent cible, le fichier source est déverrouillé avant la suppression. Ce déverrouillage signifie que le fichier source peut être modifié avant la suppression du fichier. Par conséquent, il est considéré comme dangereux de supprimer le fichier source et l'avertissement suivant s'affiche:

```
BFGTR0075W: The source file has not been deleted because it is possible that the source file was modified after the source file was transferred.
```

Dans ce cas, vérifiez que le contenu du fichier source n'est pas modifié, puis supprimez manuellement le fichier source.

Vous pouvez vérifier le statut de vos transferts dans le IBM MQ Explorer. Si des transferts apparaissent sous la forme `Stalled`, vous devrez peut-être prendre des mesures correctives car le statut bloqué indique un problème avec l'agent ou entre les deux agents impliqués dans le transfert.

### Tâches associées

«Définition d'un délai d'attente pour la récupération des transferts bloqués», à la page 343

Vous pouvez définir un délai de récupération de transfert pour les transferts de fichiers bloqués qui s'applique à tous les transferts d'un agent source. Vous pouvez également définir un délai de récupération de transfert pour un transfert individuel. Si vous définissez une durée spécifique, en secondes, pendant laquelle un agent source continue de tenter de récupérer un transfert de fichier bloqué et que le transfert échoue lorsque l'agent atteint le délai d'attente, le transfert échoue.

## Définition d'un délai d'attente pour la récupération des transferts bloqués

Vous pouvez définir un délai de récupération de transfert pour les transferts de fichiers bloqués qui s'applique à tous les transferts d'un agent source. Vous pouvez également définir un délai de récupération de transfert pour un transfert individuel. Si vous définissez une durée spécifique, en secondes, pendant laquelle un agent source continue de tenter de récupérer un transfert de fichier bloqué et que le transfert échoue lorsque l'agent atteint le délai d'attente, le transfert échoue.

### Pourquoi et quand exécuter cette tâche

Vous pouvez définir un délai d'attente de récupération de transfert qui s'applique à tous les transferts d'un agent source en ajoutant un paramètre de délai d'attente de récupération de transfert au fichier `agent.properties` de l'agent. Vous pouvez également définir un délai d'attente de reprise de transfert pour un transfert individuel à partir de la ligne de commande, avec IBM MQ Explorer ou à l'aide de tâches Apache Ant. Si une valeur de délai d'attente de reprise de transfert est définie dans le fichier

`agent.properties`, la définition du délai d'attente de reprise de transfert pour un transfert individuel remplace la valeur du fichier `agent.properties`.

Il existe trois options pour le délai d'attente de reprise de transfert:

- L'agent continue de tenter de récupérer le transfert bloqué jusqu'à ce qu'il aboutisse. Il s'agit du comportement par défaut de l'agent si le délai de récupération du transfert n'est pas défini.
- L'agent marque le transfert comme ayant échoué immédiatement après la reprise.
- L'agent continue de relancer le transfert bloqué pendant une durée spécifiée avant que le transfert ne soit marqué comme ayant échoué.

La définition du délai d'attente de récupération du transfert de fichiers est facultative. Si vous ne le définissez pas, les transferts suivent le comportement par défaut, où l'agent tente de récupérer un transfert bloqué jusqu'à ce qu'il aboutisse.

### Concepts associés

«MFT Reprise et redémarrage», à la page 343

Si votre agent ou votre gestionnaire de files d'attente n'est pas disponible pour une raison quelconque, par exemple en raison d'une panne d'alimentation ou de réseau, Managed File Transfer effectue la récupération suivante dans les scénarios suivants:

## Concepts de délai d'attente de reprise de transfert

Vous pouvez définir la durée, en secondes, pendant laquelle un agent source tente de récupérer un transfert de fichiers bloqué. Si le transfert n'aboutit pas lorsque l'agent atteint le délai d'attente pour l'intervalle entre les nouvelles tentatives, le transfert échoue.

### Priorité du délai de reprise

Une valeur de délai d'attente de reprise de transfert pour un transfert individuel spécifié via les commandes **fteCreateTransfer**, **fteCreateTemplate** ou **fteCreateMonitor**, ou à l'aide de IBM MQ Explorer, ou spécifié dans l'élément imbriqué **fte:filespec**, est prioritaire sur la valeur spécifiée pour le paramètre **transferRecoveryTimeout** dans le fichier `agent.properties` de l'agent source.

Par exemple, si la commande **fteCreateTransfer** est démarrée sans le paramètre **-rt** et la paire de valeurs, l'agent source AGENT1 recherche une valeur **transferRecoveryTimeout** dans le fichier `agent.properties` pour déterminer le comportement du délai d'attente de reprise:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

Si le paramètre **transferRecoveryTimeout** du fichier `agent.properties` n'est pas défini ou est défini sur **-1**, l'agent suit le comportement par défaut et tente de récupérer le transfert jusqu'à ce qu'il aboutisse.

Toutefois, si la commande **fteCreateTransfer** inclut le paramètre **-rt**, la valeur de ce paramètre est prioritaire sur la valeur du fichier `agent.properties` et est utilisée comme paramètre de délai de récupération pour le transfert:

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

### Compteur du délai d'attente de reprise

Le compteur de délai d'attente de récupération démarre lorsque le transfert passe à l'état de récupération. Un message de journal de transfert est publié dans SYSTEM.FTE avec la chaîne de rubrique `Log/agent_name/transfer_ID` pour indiquer que le statut de transfert est remplacé par Récupération et l'heure d'horloge de l'agent source à laquelle le statut a été modifié. Si le transfert est repris dans l'intervalle de relance défini et n'atteint pas le délai de récupération (compteur <=délai de récupération), le compteur est réinitialisé à 0, prêt à être redémarré si le transfert entre en récupération.



Si le compteur atteint la valeur maximale définie pour le délai de récupération (compteur == délai de récupération), la récupération du transfert s'arrête et l'agent source signale que le transfert a échoué. Ce type d'échec de transfert, causé par le fait que le transfert a atteint le délai de récupération, est indiqué par le code de message, `RECOVERY TIMEOUT (69)`. Un autre message de journal de transfert est publié dans `SYSTEM.FTE`, avec une chaîne de rubrique `Log/agent_name/transfer_ID`, pour indiquer que le transfert a échoué et inclut un message, le code retour et le journal des événements de l'agent source. Le journal des événements de l'agent source est mis à jour avec un message lorsque l'un des événements suivants se produit lors de la récupération:

- Lorsque le paramètre de délai d'attente de reprise est défini sur une valeur supérieure à -1, le transfert entre en phase de reprise. Le journal des événements de l'agent est mis à jour pour indiquer le début du temporisateur de récupération pour le **TransferId** et la durée pendant laquelle l'agent source attend avant de lancer le traitement du délai de récupération.
- Lorsque le transfert de récupération est repris, le journal des événements de l'agent source est mis à jour avec un nouveau message indiquant que le **TransferId** qui était en cours de récupération est repris.
- Lorsqu'un transfert de récupération a dépassé le délai d'attente, le journal des événements de l'agent source est mis à jour pour indiquer le **TransferId** qui a échoué lors de la récupération en raison du délai d'attente de récupération.

Ces messages de journal permettent aux utilisateurs (abonnés et consignateurs) d'identifier les transferts qui ont échoué en raison du délai d'attente de reprise de transfert.

Le compteur du délai de récupération est toujours au niveau de l'agent source. Toutefois, si l'agent de destination ne parvient pas à recevoir les informations de l'agent source dans les délais impartis, il peut envoyer une demande à l'agent source pour mettre le transfert en récupération. Pour un transfert où l'option de délai de récupération est définie, l'agent source démarre le compteur de délai de récupération lorsqu'il reçoit la demande de l'agent de destination.

Une gestion manuelle est toujours nécessaire pour les transferts qui n'utilisent pas l'option de délai de récupération, les transferts ayant échoué et partiellement terminés.

Pour les ensembles de transfert, où une seule demande de transfert est émise pour plusieurs fichiers et que certains des fichiers se sont terminés avec succès mais qu'un seul ne s'est terminé que partiellement, le transfert est toujours marqué comme ayant échoué car il ne s'est pas terminé comme prévu. Il se peut que l'agent source ait dépassé le délai d'attente lors du transfert du fichier partiellement terminé.

Vérifiez que l'agent de destination et le serveur de fichiers sont prêts et à l'état permettant d'accepter les transferts de fichiers.

Vous devez émettre à nouveau la demande de transfert pour l'ensemble complet, mais pour éviter les problèmes car certains des fichiers restent sur la destination à partir de la tentative de transfert initiale, vous pouvez émettre la nouvelle demande avec l'option `overwrite if existing` spécifiée. Cela permet de s'assurer que l'ensemble incomplet de fichiers de la tentative de transfert précédente est nettoyé dans le cadre du nouveau transfert, avant que les fichiers ne soient à nouveau écrits dans la destination.

Depuis IBM MQ 9.1.5, il n'est plus nécessaire de supprimer manuellement les fichiers de parties laissés sur une destination après l'échec d'une tentative de transfert initiale. Si un délai de récupération de transfert est défini pour un transfert, l'agent source fait passer le transfert à l'état `RecoveryTimedOut` si le délai de récupération de transfert est dépassé. Une fois que le transfert a été resynchronisé, l'agent de destination retire tout fichier partiel qui a été créé au cours du transfert et envoie un message indiquant que l'opération est terminée à l'agent source.

## Traces et messages

Les points de trace sont inclus à des fins de diagnostic. La valeur du délai d'attente de reprise, le début de l'intervalle entre les nouvelles tentatives, le début de la période de reprise et la réinitialisation du compteur, et si le transfert a expiré ou a échoué, sont consignés. En cas de problème ou de

comportement inattendu, vous pouvez collecter le journal de sortie et les fichiers de trace de l'agent source et les fournir à la demande du support IBM pour vous aider à résoudre les problèmes.

Les messages vous avertissent lorsque:

- Un transfert entre dans la reprise ([BFGTR0081I](#))
- Un transfert est arrêté car il a expiré depuis la récupération ([BFGSS0081E](#))
- Reprise du transfert après reprise ([BFGTR0082I](#))

### Concepts associés

«MFT Reprise et redémarrage», à la page 343

Si votre agent ou votre gestionnaire de files d'attente n'est pas disponible pour une raison quelconque, par exemple en raison d'une panne d'alimentation ou de réseau, Managed File Transfer effectue la récupération suivante dans les scénarios suivants:

## Définition du délai d'attente de récupération de transfert pour tous les transferts pour un agent source

Vous pouvez définir un délai de récupération de transfert qui s'applique à tous les transferts d'un agent source en ajoutant le paramètre **transferRecoveryTimeout** au fichier `agent.properties`.

### Pourquoi et quand exécuter cette tâche

Pour définir un délai d'attente de récupération de transfert qui s'applique à tous les transferts d'un agent source, vous ajoutez la paire de paramètres et de valeurs pour **transferRecoveryTimeout** au fichier `agent.properties`.

Il existe trois options pour le paramètre **transferRecoveryTimeout** :

**-1**

L'agent continue de tenter de récupérer le transfert bloqué jusqu'à ce que le transfert aboutisse. Cette option est équivalente au comportement par défaut de l'agent si la propriété n'est pas définie.

**0**

L'agent arrête le transfert de fichiers dès qu'il entre dans la phase de reprise.

**>0**

L'agent continue de tenter de reprendre le transfert bloqué pendant le nombre de secondes défini par l'entier positif spécifié.

Les modifications que vous apportez au fichier `agent.properties` n'entrent en vigueur qu'après le redémarrage de l'agent.

Si nécessaire, vous pouvez remplacer la valeur de délai d'attente de reprise de transfert dans le fichier `agent.properties` pour un transfert individuel. Pour plus d'informations, voir «[Définition du délai d'attente de reprise de transfert pour les transferts individuels](#)», à la page 347.

## Procédure

- Pour indiquer que l'agent continue de tenter de récupérer le transfert bloqué jusqu'à ce qu'il aboutisse, définissez une valeur de délai d'attente de récupération de transfert de `-1` comme illustré dans l'exemple suivant:

```
transferRecoveryTimeout=-1
```

- Pour indiquer que l'agent marque le transfert comme ayant échoué immédiatement après la reprise, définissez une valeur de délai d'attente de reprise de transfert de `0` comme illustré dans l'exemple suivant:

```
transferRecoveryTimeout=0
```

- Pour indiquer que l'agent effectue une nouvelle tentative de transfert bloqué pendant une période donnée avant que le transfert ne soit marqué comme ayant échoué, définissez une valeur de délai

d'attente de récupération de transfert pour la durée, en secondes, pendant laquelle vous souhaitez que l'agent effectue une nouvelle tentative.

Par exemple, la définition d'une valeur de délai d'attente de récupération de transfert de 21600 signifie que l'agent continue de tenter de récupérer le transfert pendant six heures à partir du moment où il entre dans la récupération:

```
transferRecoveryTimeout=21600
```

La valeur maximale de ce paramètre est 999999999.

## Définition du délai d'attente de reprise de transfert pour les transferts individuels

Vous pouvez définir le délai d'attente de reprise de transfert pour un transfert individuel à partir de la ligne de commande, avec IBM MQ Explorer ou à l'aide des tâches Apache Ant. Si une valeur de délai d'attente de reprise de transfert est définie dans le fichier `agent.properties`, la définition du délai d'attente de reprise de transfert pour un transfert individuel remplace la valeur définie dans le fichier `agent.properties`.

### Pourquoi et quand exécuter cette tâche

Vous pouvez définir le paramètre de délai d'attente de reprise de transfert pour un transfert individuel lorsque vous:

- Création d'un transfert à l'aide de la commande **fteCreateTransfer** ou à l'aide de IBM MQ Explorer.
- Création d'un modèle de transfert à l'aide de la commande **fteCreateTemplate** ou de IBM MQ Explorer.
- Création d'un moniteur de ressources à l'aide de la commande **fteCreateMonitor** ou de IBM MQ Explorer.
- Copie ou déplacement de fichiers à l'aide des tâches `fte: filecopy` ou `fte: filemove` Ant.

Si vous définissez une valeur de délai d'attente de reprise de transfert pour un transfert individuel, cette valeur remplace la valeur de délai d'attente de reprise de transfert définie dans le fichier `agent.properties` (voir «[Définition du délai d'attente de récupération de transfert pour tous les transferts pour un agent source](#)», à la page 346).

### Procédure

- Pour utiliser la commande **fteCreateTransfer** ou **fteCreateTemplate** afin de définir le délai d'attente de reprise de transfert, spécifiez l'option appropriée pour le paramètre **-rt** :

**-1**

L'agent continue de tenter de récupérer le transfert bloqué jusqu'à ce que le transfert aboutisse. Cette option est équivalente au comportement par défaut de l'agent si la propriété n'est pas définie.

**0**

L'agent arrête le transfert de fichiers dès qu'il entre dans la phase de reprise.

**>0**

L'agent continue de tenter de récupérer le transfert bloqué pendant la durée indiquée, en secondes.

Exemples de commande **fteCreateTransfer**

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt -1 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 0 -df C:\import\transferredfile.txt  
C:\export\originalfile.txt
```

```
fteCreateTransfer -sa AGENT1 -da AGENT2 -rt 21600 -df C:\import\transferredfile.txt
C:\export\originalfile.txt
```

## Exemples de commande **fteCreateTemplate**

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt -1 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 0 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

```
fteCreateTemplate -tn "payroll accounts monthly report template" -rt 21600 -sa PAYROLL -sm
QM_PAYROLL1 -da ACCOUNTS
-dm QM_ACCOUNTS -df C:\payroll_reports\*.xls C:\out\*.xls
```

Il n'existe pas de paramètre **-rt** pour la commande **fteCreateMonitor**. Si vous définissez le paramètre **-rt** avec la commande **fteCreateTransfer** et le paramètre **-gt**, le paramètre de délai de récupération est inclus dans le document XML avec la définition de transfert générée lorsque vous exécutez la commande **fteCreateTransfer**. Le moniteur de ressources utilise ensuite ce document XML lorsque vous exécutez la commande **fteCreateMonitor**. Dans l'exemple suivant, les détails du délai de récupération du transfert sont inclus dans le fichier `task.xml` :

```
fteCreateMonitor -ma AgentName -md C:\mqmft\monitors -mn Monitor_Name -mt task.xml -tr
"fileSize>=5MB,*.zip"
```

- Pour utiliser la page de l'assistant IBM MQ Explorer Nouveau transfert, Nouveau moniteur ou Nouveau modèle afin de définir le délai d'attente de récupération de transfert, sélectionnez l'option requise dans la zone **Délai d'attente de récupération de transfert** (secondes):

### En tant qu'agent source

Si vous sélectionnez **En tant qu'agent source**, la valeur du paramètre **transferRecoveryTimeout** du fichier `agent.properties` est utilisée si elle est définie. Dans le cas contraire, le comportement par défaut du délai d'attente de reprise de transfert est appliqué.

### Zone de liste numérique

Si vous entrez une durée en secondes dans la zone de liste numérique, l'agent continue de tenter de récupérer le transfert bloqué pendant la durée spécifiée.

### Aucun

Si vous sélectionnez **Aucun**, aucun délai de récupération de transfert n'est défini et l'agent continue de tenter de récupérer le transfert bloqué jusqu'à ce que le transfert aboutisse.

- Pour définir le délai de récupération à l'aide des tâches Ant, inclure l'option et la valeur **transferRecoveryTimeout**, avec les éléments **fte:filecopy** ou **fte:filemove** pour le déplacement ou la copie de fichiers, par exemple:

### Exemple pour **fte:filecopy**

```
<fte:filecopy cmdqmq="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="copy.result" transferRecoveryTimeout="0">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filecopy>
```

### Exemple pour **fte:filemove**

```
<fte:filemove cmdqmq="qm0@localhost@1414@SYSTEM.DEF.SVRCONN"
src="agent1@qm1" dst="agent2@qm2"
rcproperty="move.result" transferRecoveryTimeout="21600">
  <fte:filespec srcfilespec="/home/fteuser1/file.bin" dstfile="/home/fteuser2/file.bin"/>
</fte:filemove>
```

## Concepts associés

[«MFT Reprise et redémarrage», à la page 343](#)

Si votre agent ou votre gestionnaire de files d'attente n'est pas disponible pour une raison quelconque, par exemple en raison d'une panne d'alimentation ou de réseau, Managed File Transfer effectue la récupération suivante dans les scénarios suivants:

## Tâches associées

[«Démarrage d'un nouveau transfert de fichiers», à la page 228](#)

Vous pouvez démarrer un nouveau transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande et vous pouvez choisir de transférer un seul fichier ou plusieurs fichiers dans un groupe.

[«Création d'un modèle de transfert de fichiers à l'aide de IBM MQ Explorer», à la page 274](#)

Vous pouvez créer un modèle de transfert de fichier à partir de IBM MQ Explorer ou de la ligne de commande. Vous pouvez ensuite utiliser ce modèle pour créer de nouveaux transferts de fichiers à l'aide des détails du modèle ou soumettre le modèle pour démarrer le transfert de fichiers.

[«Surveillance des ressources MFT», à la page 239](#)

Vous pouvez surveiller les ressources Managed File Transfer ; par exemple, une file d'attente ou un répertoire. Lorsqu'une condition est satisfaite sur cette ressource, le moniteur de ressources démarre une tâche, telle qu'un transfert de fichier. Vous pouvez créer un moniteur de ressources à l'aide de la commande **fteCreateMonitor** ou de la vue **Moniteurs** dans le plug-in Managed File Transfer pour IBM MQ Explorer.

[«Affichage du statut des transferts de fichiers dans le journal de transfert», à la page 237](#)

Vous pouvez afficher les détails des transferts de fichiers à l'aide du **Journal de transfert** dans IBM MQ Explorer. Il peut s'agir de transferts démarrés à partir de la ligne de commande ou du IBM MQ Explorer. Vous pouvez également personnaliser les éléments affichés dans le **Journal de transfert**.

## Référence associée

[Le fichier MFT agent.properties](#)

**fteCreateTransfer**: démarrer un nouveau transfert de fichier

**fteCreateTemplate**: créer un nouveau modèle de transfert de fichiers

**fteCreateMonitor**: créer un moniteur de ressources MFT

fte: tâche Ant filecopy

fte: tâche Ant filemove

Windows

Linux

AIX

## Administration de MQ Telemetry

MQ Telemetry est administré à l'aide de IBM MQ Explorer ou d'une ligne de commande. Utilisez l'explorateur pour configurer les canaux de télémétrie, contrôler le service de télémétrie et surveiller les clients MQTT qui sont connectés à IBM MQ. Configurez la sécurité de MQ Telemetry à l'aide de JAAS, de TLS et du gestionnaire des droits d'accès aux objets IBM MQ .

### Administration à l'aide de IBM MQ Explorer

Utilisez l'explorateur pour configurer les canaux de télémétrie, contrôler le service de télémétrie et surveiller les clients MQTT qui sont connectés à IBM MQ. Configurez la sécurité de MQ Telemetry à l'aide de JAAS, de TLS et du gestionnaire des droits d'accès aux objets IBM MQ .

### Administration à l'aide de la ligne de commande

MQ Telemetry peut être entièrement administré à partir de la ligne de commande à l'aide des commandes [MQSC](#).

La documentation MQ Telemetry contient également des exemples de script qui illustrent l'utilisation de base de l'application client IBM MQ Telemetry Transport v3 .

Lisez et comprenez les exemples dans les [exemples de programmes IBM MQ Telemetry Transport](#) avant de les utiliser.

## Concepts associés

[MQ Telemetry](#)

## Référence associée

Propriétés de MQXR

Linux

AIX

# Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Linux et AIX

Procédez comme suit pour configurer MQ Telemetry manuellement. Si vous n'avez besoin que d'une configuration simple qui utilise l'ID utilisateur invité, vous pouvez exécuter l'assistant de support MQ Telemetry dans IBM MQ Explorer.

## Avant de commencer

Si vous n'avez besoin que d'une configuration simple, envisagez d'utiliser le support MQ Telemetry dans IBM MQ Explorer. Cette prise en charge inclut un assistant et un exemple de procédure de commande `sampleMQM`. Ces ressources définissent une configuration initiale à l'aide de l'ID utilisateur invité. Voir [Vérification de l'installation de MQ Telemetry à l'aide des exemples de programme IBM MQ Explorer](#) et [IBM MQ Telemetry Transport](#).

Si vous avez besoin d'une configuration plus complexe qui utilise une méthode d'authentification différente, suivez les étapes de cette tâche. Commencez par les étapes initiales suivantes:

1. Pour plus d'informations sur l'installation de IBM MQ et de la fonction MQ Telemetry, voir [Remarques relatives à l'installation d' MQ Telemetry](#).
2. Créez et démarrez un gestionnaire de files d'attente. Le gestionnaire de files d'attente est appelé *qMgr* dans cette tâche.
3. Dans le cadre de cette tâche, vous configurez le service de télémétrie (MQXR). Les paramètres de propriété MQXR sont stockés dans un fichier de propriétés spécifique à la plateforme: `mqxr_win.properties`. En règle générale, vous n'avez pas besoin d'éditer directement le fichier de propriétés MQXR, car la plupart des paramètres peuvent être configurés via des commandes d'administration MQSC ou IBM MQ Explorer. Si vous décidez d'éditer le fichier directement, arrêtez le gestionnaire de files d'attente avant d'effectuer vos modifications. Voir [MQXR properties](#).

## Pourquoi et quand exécuter cette tâche

Suivez les étapes de cette tâche pour configurer MQ Telemetry manuellement à l'aide de différents schémas d'autorisation.

## Procédure

1. Ouvrez une fenêtre de commande dans le répertoire des exemples de télémétrie.

Le répertoire des exemples de télémétrie est `/opt/mqm/mqxr/samples`.

2. Créez la file d'attente de transmission de télémétrie.

Si `SYSTEM.MQTT.TRANSMIT.QUEUE` n'existe pas, il est créé automatiquement lorsque le service de télémétrie (MQXR) est démarré pour la première fois et défini pour utiliser l'ID utilisateur invité. Toutefois, cette tâche configure MQ Telemetry pour utiliser un schéma d'autorisation différent. Pour cette tâche, vous créez `SYSTEM.MQTT.TRANSMIT.QUEUE` et configurez l'accès à ce dernier avant de démarrer le service de télémétrie (MQXR).

Exécutez ensuite la commande suivante :

```
echo "DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000)" | runmqsc qMgr
```

3. Définissez la file d'attente de transmission par défaut.

Il est plus facile d'envoyer des messages directement aux clients MQTT si `SYSTEM.MQTT.TRANSMIT.QUEUE` est la file d'attente de transmission par défaut. Sinon, vous devez

ajouter une définition de file d'attente éloignée pour chaque client qui reçoit des messages IBM MQ ; voir «Envoi direct d'un message à un client», à la page 355. Notez que la modification de la file d'attente de transmission par défaut peut interférer avec votre configuration existante.

Lorsque le service de télémétrie (MQXR) est démarré pour la première fois, il ne définit pas SYSTEM.MQTT.TRANSMIT.QUEUE comme file d'attente de transmission par défaut pour le gestionnaire de files d'attente. Pour configurer ce paramètre, vous devez modifier la propriété de file d'attente de transmission par défaut. Pour ce faire, utilisez la IBM MQ Explorer ou exécutez la commande suivante:

```
echo "ALTER QMGR DEFEXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') " | runmqsc qMgr
```

4. Suivez une procédure dans «Autorisation des clients MQTT à accéder aux objets IBM MQ», à la page 358 pour créer un ou plusieurs ID utilisateur. Les ID utilisateur ont le droit de publier, de s'abonner et d'envoyer des publications aux clients MQTT .
5. Editez le fichier `installMQXRService_unix.mqsc` pour configurer le fichier de clés utilisé pour chiffrer la phrase passe pour les canaux TLS MQTT:

- a) Ouvrez le fichier `WMQ program installation directory/mqxr/samples/installMQXRService_unix.mqsc`.

- b) Recherchez la ligne qui inclut le paramètre **STARTARGET** éditez l'option **-sf** pour spécifier l'emplacement de votre fichier de clés de données d'identification.

Par défaut, le fichier `installMQXRService_unix.mqsc` utilise un fichier de clés par défaut nommé [DEFAULT]. Le fichier de clés par défaut étant le même pour toutes les installations IBM MQ , vous devez fournir un fichier de clés unique à votre installation lorsque vous chiffrez des phrases passe.

Voir aussi l'exemple de code dans «Création du SYSTEM.MQXR.SERVICE», à la page 351.

6. Installez le service de télémétrie (MQXR) en exécutant la commande suivante:

```
cat /opt/<install_dir>/mqxr/samples/installMQXRService_unix.mqsc | runmqsc queue_manager
```

7. Démarrez le service.

```
echo "START SERVICE(SYSTEM.MQXR.SERVICE) " | runmqsc qMgr
```

Le service de télémétrie (MQXR) est démarré automatiquement au démarrage du gestionnaire de files d'attente. Il est démarré manuellement dans cette tâche, car le gestionnaire de files d'attente est déjà en cours d'exécution.

8. A l'aide de IBM MQ Explorer, configurez les canaux de télémétrie pour accepter les connexions des clients MQTT .

Les canaux de télémétrie doivent être configurés de sorte que leurs identités soient l'un des ID utilisateur définis à l'étape «4», à la page 351.

Voir aussi [DEFINE CHANNEL \(MQTT\)](#).

9. Vérifiez la configuration en exécutant l'exemple de client.

Pour que l'exemple de client puisse utiliser votre canal de télémétrie, le canal doit autoriser le client à publier, s'abonner et recevoir des publications. L'exemple de client se connecte au canal de télémétrie sur le port 1883 par défaut. Voir aussi [exemples de programmes IBM MQ Telemetry Transport](#).

## Création du SYSTEM.MQXR.SERVICE

Utilisez la commande **runMQXRService** pour créer le SYSTEM.MQXR.SERVICE.

```
DEF      SERVICE(SYSTEM.MQXR.SERVICE) +  
CONTROL(QMGR) +  
DESCR('Manages clients using MQXR protocols such as MQTT') +
```

```
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+/mqxr/bin/runMQXRService.sh') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+" -g "+MQ_DATA_PATH+" -sf "/home/keyFileLocation/
keyFile.txt"') +
STOPCMD('+MQ_INSTALL_PATH+/mqxr/bin/endMQXRService.sh') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+/mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+/mqxr.stderr')
```

## Windows Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Windows

Procédez comme suit pour configurer MQ Telemetry manuellement. Si vous n'avez besoin que d'une configuration simple qui utilise l'ID utilisateur invité, vous pouvez exécuter l'assistant de support MQ Telemetry dans IBM MQ Explorer.

### Avant de commencer

Si vous n'avez besoin que d'une configuration simple, envisagez d'utiliser le support MQ Telemetry dans IBM MQ Explorer. Cette prise en charge inclut un assistant et un exemple de procédure de commande `sampleMQM`. Ces ressources définissent une configuration initiale à l'aide de l'ID utilisateur invité. Voir [Vérification de l'installation de MQ Telemetry à l'aide des exemples de programme IBM MQ Explorer](#) et [IBM MQ Telemetry Transport](#).

Si vous avez besoin d'une configuration plus complexe qui utilise une méthode d'authentification différente, suivez les étapes de cette tâche. Commencez par les étapes initiales suivantes:

1. Pour plus d'informations sur l'installation de IBM MQ et de la fonction MQ Telemetry, voir [Remarques relatives à l'installation d' MQ Telemetry](#).
2. Créez et démarrez un gestionnaire de files d'attente. Le gestionnaire de files d'attente est appelé *qMgr* dans cette tâche.
3. Dans le cadre de cette tâche, vous configurez le service de télémétrie (MQXR). Les paramètres de propriété MQXR sont stockés dans un fichier de propriétés spécifique à la plateforme: `mqxr_win.properties`. En règle générale, vous n'avez pas besoin d'éditer directement le fichier de propriétés MQXR, car la plupart des paramètres peuvent être configurés via des commandes d'administration MQSC ou IBM MQ Explorer. Si vous décidez d'éditer le fichier directement, arrêtez le gestionnaire de files d'attente avant d'effectuer vos modifications. Voir [MQXR properties](#).

### Pourquoi et quand exécuter cette tâche

Suivez les étapes de cette tâche pour configurer MQ Telemetry manuellement à l'aide de différents schémas d'autorisation.

### Procédure

1. Ouvrez une fenêtre de commande dans le répertoire des exemples de télémétrie.

Le répertoire des exemples de télémétrie est *WMQ program installation directory\mqxr\samples*.

2. Créez la file d'attente de transmission de télémétrie.

Si `SYSTEM.MQTT.TRANSMIT.QUEUE` n'existe pas, il est créé automatiquement lorsque le service de télémétrie (MQXR) est démarré pour la première fois et défini pour utiliser l'ID utilisateur invité. Toutefois, cette tâche configure MQ Telemetry pour utiliser un schéma d'autorisation différent. Pour cette tâche, vous créez `SYSTEM.MQTT.TRANSMIT.QUEUE` et configurez l'accès à ce dernier avant de démarrer le service de télémétrie (MQXR).

Exécutez ensuite la commande suivante :

```
echo DEFINE QLOCAL('SYSTEM.MQTT.TRANSMIT.QUEUE') USAGE(XMITQ) MAXDEPTH(100000) | runmqsc qMgr
```



### 3. Définissez la file d'attente de transmission par défaut.

Il est plus facile d'envoyer des messages directement aux clients MQTT si `SYSTEM.MQTT.TRANSMIT.QUEUE` est la file d'attente de transmission par défaut. Sinon, vous devez ajouter une définition de file d'attente éloignée pour chaque client qui reçoit des messages IBM MQ ; voir «[Envoi direct d'un message à un client](#)», à la [page 355](#). Notez que la modification de la file d'attente de transmission par défaut peut interférer avec votre configuration existante.

Lorsque le service de télémétrie (MQXR) est démarré pour la première fois, il ne définit pas `SYSTEM.MQTT.TRANSMIT.QUEUE` comme file d'attente de transmission par défaut pour le gestionnaire de files d'attente. Pour configurer ce paramètre, vous devez modifier la propriété de file d'attente de transmission par défaut. Pour ce faire, utilisez la IBM MQ Explorer ou exécutez la commande suivante:

```
echo ALTER QMGR DEFXMITQ('SYSTEM.MQTT.TRANSMIT.QUEUE') | runmqsc qMgr
```

### 4. Suivez une procédure dans «[Autorisation des clients MQTT à accéder aux objets IBM MQ](#)», à la [page 358](#) pour créer un ou plusieurs ID utilisateur. Les ID utilisateur ont le droit de publier, de s'abonner et d'envoyer des publications aux clients MQTT .

### 5. Editez le fichier `installMQXRService_win.mqsc` pour configurer le fichier de clés utilisé pour chiffrer la phrase passe pour les canaux TLS MQTT:

a) Ouvrez le fichier *WMQ program installation*

`directory\mqxr\samples\installMQXRService_win.mqsc`.

b) Recherchez la ligne qui inclut le paramètre **STARTARGET** éditez l'option **-sf** pour spécifier l'emplacement de votre fichier de clés de données d'identification.

Par défaut, le fichier `installMQXRService_win.mqsc` utilise un fichier de clés par défaut nommé [DEFAULT]. Le fichier de clés par défaut étant le même pour toutes les installations IBM MQ , vous devez fournir un fichier de clés unique à votre installation lorsque vous chiffrez des phrases passe.

Voir aussi l'exemple de code dans «[Création SYSTEM.MQXR.SERVICE](#)», à la [page 354](#).

### 6. Installez le service de télémétrie (MQXR) en exécutant la commande suivante:

```
type installMQXRService_win.mqsc | runmqsc qMgr
```

### 7. Démarrez le service.

```
echo START SERVICE(SYSTEM.MQXR.SERVICE) | runmqsc qMgr
```

Le service de télémétrie (MQXR) est démarré automatiquement au démarrage du gestionnaire de files d'attente. Il est démarré manuellement dans cette tâche, car le gestionnaire de files d'attente est déjà en cours d'exécution.

### 8. A l'aide de IBM MQ Explorer, configurez les canaux de télémétrie pour accepter les connexions des clients MQTT .

Les canaux de télémétrie doivent être configurés de sorte que leurs identités soient l'un des ID utilisateur définis à l'étape «4», à la [page 353](#).

Voir aussi [DEFINE CHANNEL \(MQTT\)](#).

### 9. Vérifiez la configuration en exécutant l'exemple de client.

Pour que l'exemple de client puisse utiliser votre canal de télémétrie, le canal doit autoriser le client à publier, s'abonner et recevoir des publications. L'exemple de client se connecte au canal de télémétrie sur le port 1883 par défaut. Voir aussi [exemples de programmes IBM MQ Telemetry Transport](#).

## Création SYSTEM.MQXR.SERVICE

Utilisez la commande **runMQXRService** pour créer le SYSTEM.MQXR.SERVICE.

```
DEF SERVICE(SYSTEM.MQXR.SERVICE) +
CONTROL(QMGR) +
DESCR('Manages clients using MQXR protocols such as MQTT') +
SERVTYPE(SERVER) +
STARTCMD('+MQ_INSTALL_PATH+\mqxr\bin\runMQXRService.bat') +
STARTARG('-m +QMNAME+ -d "+MQ_Q_MGR_DATA_PATH+\" -g "+MQ_DATA_PATH+\" -sf
"c:\keyFileLocation\keyFile.txt"') +
STOPCMD('+MQ_INSTALL_PATH+\mqxr\bin\endMQXRService.bat') +
STOPARG('-m +QMNAME+') +
STDOUT('+MQ_Q_MGR_DATA_PATH+\mqxr.stdout') +
STDERR('+MQ_Q_MGR_DATA_PATH+\mqxr.stderr')
```

Windows

Linux

AIX

## Configuration de la mise en file d'attente répartie pour l'envoi de messages aux clients MQTT

Les applications IBM MQ peuvent envoyer des messages aux clients MQTT v3 en les publiant dans un abonnement créé par un client ou en envoyant un message directement. Quelle que soit la méthode utilisée, le message est placé sur SYSTEM.MQTT.TRANSMIT.QUEUE et envoyé au client par le service de télémétrie (MQXR). Il existe plusieurs façons de placer un message sur SYSTEM.MQTT.TRANSMIT.QUEUE.

### Publication d'un message en réponse à un abonnement client MQTT

Le service de télémétrie (MQXR) crée un abonnement pour le compte du client MQTT. Le client est la destination des publications qui correspondent à l'abonnement envoyé par le client. Les services de télémétrie réachemine les publications correspondantes vers le client.

Un client MQTT est connecté à IBM MQ en tant que gestionnaire de files d'attente, son nom de gestionnaire de files d'attente étant défini sur *ClientIdentifier*. La destination des publications à envoyer au client est une file d'attente de transmission, SYSTEM.MQTT.TRANSMIT.QUEUE. Le service de télémétrie transmet les messages sur SYSTEM.MQTT.TRANSMIT.QUEUE aux clients MQTT, en utilisant le nom du gestionnaire de files d'attente cible comme clé d'un client spécifique.

Le service de télémétrie (MQXR) ouvre la file d'attente de transmission en utilisant *ClientIdentifier* comme nom de gestionnaire de files d'attente. Le service de télémétrie (MQXR) transmet le descripteur d'objet de la file d'attente à l'appel MQSUB pour transmettre les publications qui correspondent à l'abonnement du client. Dans la résolution de nom d'objet, *ClientIdentifier* est créé en tant que nom de gestionnaire de files d'attente éloignées et la file d'attente de transmission doit être résolue en SYSTEM.MQTT.TRANSMIT.QUEUE. A l'aide de la résolution de nom d'objet IBM MQ standard, *ClientIdentifier* est résolu comme suit ; voir [Tableau 16](#), à la page 355.

1. *ClientIdentifier* ne correspond à rien.

*ClientIdentifier* est un nom de gestionnaire de files d'attente éloignées. Il ne correspond pas au nom du gestionnaire de files d'attente local, à un alias de gestionnaire de files d'attente ou à un nom de file d'attente de transmission.

Le nom de la file d'attente n'est pas défini. Actuellement, le service de télémétrie (MQXR) définit SYSTEM.MQTT.PUBLICATION.QUEUE comme nom de la file d'attente. Un client MQTT v3 ne prenant pas en charge les files d'attente, le nom de la file d'attente résolue est ignoré par le client. La propriété du gestionnaire de files d'attente local, File d'attente de transmission par défaut, doit être définie sur SYSTEM.MQTT.TRANSMIT.QUEUE, de sorte que la publication soit placée sur SYSTEM.MQTT.TRANSMIT.QUEUE pour être envoyée au client.

2. *ClientIdentifier* correspond à un alias de gestionnaire de files d'attente nommé *ClientIdentifier*.

*ClientIdentifier* est un nom de gestionnaire de files d'attente éloignées. Il correspond au nom d'un alias de gestionnaire de files d'attente.

L'alias de gestionnaire de files d'attente doit être défini avec *ClientIdentif*ier comme nom de gestionnaire de files d'attente éloignées.

En définissant le nom de la file d'attente de transmission dans la définition d'alias de gestionnaire de files d'attente, il n'est pas nécessaire que la transmission par défaut soit définie sur SYSTEM.MQTT.TRANSMIT.QUEUE.

Tableau 16. Résolution de nom d'un alias de gestionnaire de files d'attente MQTT

<i>ClientIdentif</i> ier	Entrée		Sortie		
	Nom gest. de files	Nom de la file d'attente	Nom gest. de files	Nom de la file d'attente	File d'attente de transmission
Ne correspond à rien	<i>ClientIdentif</i> ier	<i>non défini</i>	<i>ClientIdentif</i> ier	<i>non défini</i>	File d'attente de transmission par défaut.  SYSTEM.MQTT.TRANSMIT.QUEUE
Correspond à un alias de gestionnaire de files d'attente nommé <i>ClientIdentif</i> ier	<i>ClientIdentif</i> ier	<i>non défini</i>	<i>ClientIdentif</i> ier	<i>non défini</i>	SYSTEM.MQTT.TRANSMIT.QUEUE

Pour plus d'informations sur la résolution de nom, voir [Résolution de nom](#).

Tout programme IBM MQ peut publier dans la même rubrique. La publication est envoyée à ses abonnés, y compris les clients MQTT v3 qui ont un abonnement à la rubrique.

Si une rubrique d'administration est créée dans un cluster, avec l'attribut CLUSTER(*clusterName*), toute application du cluster peut être publiée sur le client ; par exemple:

```
echo DEFINE TOPIC('MQTTExamples') TOPICSTR('MQTT Examples') CLUSTER(MQTT) REPLACE | runmqsc qMgr
```

**Remarque :** N'attribuez pas à SYSTEM.MQTT.TRANSMIT.QUEUE un attribut de cluster.

Les abonnés du client MQTT et les diffuseurs de publications peuvent se connecter à différents gestionnaires de files d'attente. Les abonnés et les diffuseurs peuvent faire partie du même cluster ou être connectés par une hiérarchie de publication / abonnement. La publication est distribuée par le diffuseur de publications à l'abonné à l'aide de IBM MQ.

## Envoi direct d'un message à un client

Alternative à un client qui crée un abonnement et qui reçoit une publication correspondant à la rubrique d'abonnement, qui envoie directement un message à un client MQTT v3. Les applications client MQTT V3 ne peuvent pas envoyer de messages directement, contrairement à d'autres applications, telles que les applications IBM MQ.

L'application IBM MQ doit connaître *ClientIdentif*ier du client MQTT v3. Comme les clients MQTT v3 n'ont pas de files d'attente, le nom de la file d'attente cible est transmis à la méthode `messageArrived` du client d'application MQTT v3 en tant que nom de rubrique. Par exemple, dans un programme MQI, créez un descripteur d'objet avec le client comme `ObjectQmgrNom`:

```
MQOD.ObjectQmgrName = ClientIdentif ;
MQOD.ObjectName = name ;
```

Si l'application est écrite à l'aide de JMS, créez une destination point à point, par exemple:

#### JM 3.0

```
jakarta.jms.Destination jmsDestination =
(jakarta.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifieur/name");
```

#### JMS 2.0

```
javax.jms.Destination jmsDestination =
(javax.jms.Destination)jmsFactory.createQueue
("queue://ClientIdentifieur/name");
```

Pour envoyer un message non sollicité à un client MQTT, utilisez une définition de file d'attente éloignée. Le nom du gestionnaire de files d'attente éloignées doit être résolu en `ClientIdentifieur` du client. La file d'attente de transmission doit être résolue en `SYSTEM.MQTT.TRANSMIT.QUEUE`; voir [Tableau 17](#), à la page 356. Le nom de la file d'attente éloignée peut être n'importe quel nom. Le client le reçoit sous forme de chaîne de rubrique.

Entrée		Sortie		
Nom de la file d'attente	Nom gest. de files	Nom de la file d'attente	Nom gest. de files	File d'attente de transmission
Nom de la définition de file d'attente éloignée	Nom de gestionnaire de files d'attente local ou vide	Nom de file d'attente éloignée utilisé comme chaîne de rubrique	<code>ClientIdentifieur</code>	<code>SYSTEM.MQTT.TRANSMIT.QUEUE</code>

Si le client est connecté, le message est envoyé directement au client MQTT, qui appelle la méthode `messageArrived`; voir [MéthodemessageArrived](#).

Si le client s'est déconnecté avec une session persistante, le message est stocké dans `SYSTEM.MQTT.TRANSMIT.QUEUE`; voir [MQTT sessions sans état et avec état](#). Il est transmis au client lorsque ce dernier se reconnecte à nouveau à la session.

Si vous envoyez un message non persistant, il est envoyé au client avec la qualité de service au plus une fois, `QoS=0`. Si vous envoyez un message persistant directement à un client, par défaut, il est envoyé avec la qualité de service une seule fois, `QoS=2`. Comme le client n'a peut-être pas de mécanisme de persistance, il peut réduire la qualité de service qu'il accepte pour les messages envoyés directement. Pour réduire la qualité de service des messages envoyés directement à un client, abonnez-vous à la rubrique `DEFAULT.QoS`. Indiquez la qualité de service maximale que le client peut prendre en charge.

## Windows Linux AIX MQTT Identification, autorisation et authentification du client

Le service de télémétrie (MQXR) publie des rubriques IBM MQ ou s'y abonne pour le compte de clients MQTT, à l'aide de canaux MQTT. L'administrateur IBM MQ configure l'identité de canal MQTT qui est utilisée pour l'autorisation IBM MQ. L'administrateur peut définir une identité commune pour le canal ou utiliser le Nom d'utilisateur ou l'Identificateur client d'un client connecté au canal.

Le service de télémétrie (MQXR) peut authentifier le client à l'aide du Nom d'utilisateur fourni par le client ou à l'aide d'un certificat client. Le Nom d'utilisateur est authentifié à l'aide d'un mot de passe fourni par le client.

En résumé : l'identification client est la sélection de l'identité du client. En fonction du contexte, le client est identifié par l'Identificateur client, le Nom d'utilisateur, une identité client commune créée par l'administrateur ou un certificat client. L'identificateur client utilisé pour le contrôle d'authenticité ne doit pas être le même que celui utilisé pour l'autorisation.

Les programmes client MQTT définissent le nom d'utilisateur et le mot de passe qui sont envoyés au serveur à l'aide d'un canal MQTT . Ils peuvent également définir les propriétés TLS requises pour chiffrer et authentifier la connexion. L'administrateur décide si le canal MQTT doit être authentifié et comment l'authentifier.

Pour autoriser un client MQTT à accéder à des objets IBM MQ , autorisez `ClientIdentifier` ou `Username` du client ou autorisez une identité client commune. Pour autoriser un client à se connecter à IBM MQ, vous devez authentifier le nom d'utilisateur ou utiliser un certificat client. Configurez JAAS pour authentifier le nom d'utilisateur et configurez TLS pour authentifier un certificat client.

Si vous définissez un mot de passe sur le client, chiffrez la connexion à l'aide de VPN ou configurez le canal MQTT pour qu'il utilise TLS afin de conserver le mot de passe privé.

Il est difficile de gérer des certificats client. Pour cette raison, si les risques associés à l'authentification par mot de passe sont acceptables, ce type d'authentification est souvent utilisé pour l'authentification des clients.

S'il existe un moyen sûr de gérer et de stocker le certificat client, il est possible de faire confiance à l'authentification par certificat. Toutefois, il arrive rarement que les certificats puissent être gérés de manière sécurisée dans les types d'environnements dans lesquels la télémétrie est utilisée. L'authentification des dispositifs utilisant les certificats client est complétée par l'authentification des mots de passe client au niveau du serveur. En raison de la complexité supplémentaire, l'utilisation du certificat client est limitée aux applications hautement sensibles. L'utilisation de deux formulaires d'authentification est appelée authentification à deux facteurs. Vous devez connaître l'un des facteurs, tels que le mot de passe, et disposer de l'autre, un certificat, par exemple.

Dans une application sensible, telle qu'un appareil utilisant un code confidentiel, l'appareil est verrouillé pendant sa fabrication pour empêcher toute contrefaçon avec le matériel et le logiciel internes. Un certificat client digne de confiance, limité dans le temps, est copié dans le dispositif. Le dispositif est déployé à l'emplacement où il doit être utilisé. Une authentification supplémentaire est effectuée chaque fois que le dispositif est utilisé, soit à l'aide d'un mot de passe, soit en utilisant un autre certificat à partir d'une carte à puce.

## Windows Linux AIX MQTT Identité et autorisation du client

Utilisez l'ID client, `Username`, ou une identité de client commune pour l'autorisation d'accès aux objets IBM MQ .

L'administrateur IBM MQ dispose de trois options pour sélectionner l'identité du canal MQTT . L'administrateur fait le choix lors de la définition ou de la modification du canal MQTT utilisé par le client. L'identité est utilisée pour autoriser l'accès aux rubriques IBM MQ. Le choix se fait dans l'ordre suivant:

1. L'ID client (voir [USECLNTID](#) ).
2. Identité fournie par l'administrateur pour le canal ( `MCAUSER` du canal. Voir [MCAUSER](#) ).
3. Si aucun des choix précédents ne s'applique, le Nom d'utilisateur transmis par le client MQTT ( Nom d'utilisateur est un attribut de la classe `MqttConnectOptions` . Il doit être défini avant que le client ne se connecte au service. Sa valeur par défaut est null).

**Éviter les incidents :** L'identité choisie par ce processus est ensuite désignée, par exemple par la commande `DISPLAY CHSTATUS (MQTT)`, comme `MCAUSER` du client. Sachez qu'il ne s'agit pas nécessairement de la même identité que le `MCAUSER` du canal auquel il est fait référence dans le choix (2).

La commande IBM MQ `setmqaut` permet de sélectionner les objets et les actions autorisés à être utilisés par l'identité associée au canal MQTT . Par exemple, le code suivant autorise une identité de canal `MQTTClient`, fournie par l'administrateur du gestionnaire de files d'attente `QM1`:

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

## IBM MQ

Procédez comme suit pour autoriser les clients MQTT à publier des objets IBM MQ et à s'y abonner. Ces étapes décrivent quatre modèles de contrôle d'accès différents.

### Avant de commencer

Les clients MQTT sont autorisés à accéder aux objets dans IBM MQ en se faisant affecter une identité lorsqu'ils se connectent à un canal de télémétrie. L'administrateur IBM MQ configure le canal de télémétrie à l'aide de IBM MQ Explorer pour donner à un client l'un des trois types d'identité suivants:

1. `ClientIdentifier`
2. Nom d'utilisateur
3. Nom que l'administrateur affecte au canal.

Quel que soit le type utilisé, l'identité doit être définie dans IBM MQ en tant que principal par le service d'autorisation installé. Le service d'autorisation par défaut sous Windows ou Linux est appelé Object Authority Manager (OAM). Si vous utilisez la méthode d'accès aux objets (OAM), l'identité doit être définie en tant qu'ID utilisateur.

Utilisez l'identité pour accorder à un client ou à une collection de clients le droit de publier ou de s'abonner à des rubriques définies dans IBM MQ. Si un client MQTT s'est abonné à une rubrique, utilisez l'identité pour lui donner l'autorisation de recevoir les publications résultantes.

Il est difficile de gérer un système avec des dizaines de milliers de clients MQTT, chacun nécessitant des droits d'accès individuels. Une solution consiste à définir des identités communes et à associer des clients MQTT individuels à l'une des identités communes. Définissez autant d'identités communes que nécessaire pour définir différentes combinaisons de droits. Une autre solution consiste à écrire votre propre service d'autorisation qui peut traiter plus facilement avec des milliers d'utilisateurs que le système d'exploitation.

Vous pouvez combiner les clients MQTT en identités communes de deux manières, à l'aide de la méthode d'accès aux objets (OAM):

1. Définissez plusieurs canaux de télémétrie, chacun avec un ID utilisateur différent que l'administrateur alloue à l'aide de IBM MQ Explorer. Les clients qui se connectent à l'aide de numéros de port TCP/IP différents sont associés à des canaux de télémétrie différents et se voient attribuer des identités différentes.
2. Définissez un canal de télémétrie unique, mais chaque client doit sélectionner un nom d'utilisateur dans un petit ensemble d'ID utilisateur. L'administrateur configure le canal de télémétrie pour sélectionner le nom d'utilisateur du client comme identité.

Dans cette tâche, l'identité du canal de télémétrie est appelée *mqttUser*, quel que soit le mode de définition. Si les collections de clients utilisent des identités différentes, utilisez plusieurs *mqttUsers*, un pour chaque collection de clients. Comme la tâche utilise la méthode d'accès aux objets (OAM), chaque *mqttUser* doit être un ID utilisateur.

### Pourquoi et quand exécuter cette tâche

Dans cette tâche, vous avez le choix entre quatre modèles de contrôle d'accès que vous pouvez adapter à des exigences spécifiques. Les modèles diffèrent dans leur granularité du contrôle d'accès.

- [«Aucun contrôle d'accès», à la page 359](#)
- [«Contrôle d'accès à granularité grossière», à la page 359](#)
- [«Contrôle d'accès à granularité moyenne», à la page 359](#)
- [«Contrôle d'accès à granularité fine», à la page 359](#)

Le résultat de ces modèles est d'affecter à *mqttUsers* des ensembles de droits d'accès pour la publication et l'abonnement à IBM MQ, et de recevoir des publications de IBM MQ.

### *Aucun contrôle d'accès*

Les clients MQTT disposent des droits d'administration IBM MQ et peuvent effectuer n'importe quelle action sur n'importe quel objet.

## Procédure

1. Créez un ID utilisateur *mqttUser* pour agir en tant qu'identité de tous les clients MQTT .
2. Ajoutez *mqttUser* au groupe *mqm* ; voir [Ajout d'un utilisateur à un groupe sur Windows](#) ou [Création et gestion de groupes sur Linux](#)

### *Contrôle d'accès à granularité grossière*

Les clients MQTT sont autorisés à publier et à s'abonner et à envoyer des messages aux clients MQTT . Ils n'ont pas le droit d'effectuer d'autres actions ou d'accéder à d'autres objets.

## Procédure

1. Créez un ID utilisateur *mqttUser* pour agir en tant qu'identité de tous les clients MQTT .
2. Autorisez *mqttUser* à publier et à s'abonner à toutes les rubriques et à envoyer des publications aux clients MQTT .

```
setmqaut -m qMgr -t topic -n SYSTEM.BASE.TOPIC -p mqttUser -all +pub +sub  
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqttUser -all +put
```

### *Contrôle d'accès à granularité moyenne*

Les clients MQTT sont divisés en différents groupes pour publier et s'abonner à différents ensembles de rubriques et pour envoyer des messages aux clients MQTT .

## Procédure

1. Créez plusieurs ID utilisateur, *mqttUser* et plusieurs rubriques d'administration dans l'arborescence de rubriques de publication / abonnement.
2. Autoriser différents *mqttUsers* à accéder à des rubriques différentes.

```
setmqaut -m qMgr -t topic -n topic1 -p mqttUserA -all +pub +sub  
setmqaut -m qMgr -t topic -n topic2 -p mqttUserB -all +pub +sub
```

3. Créez un groupe *mqtt* ajoutez-y tous les *mqttUsers* .
4. Autorisez *mqtt* à envoyer des rubriques aux clients MQTT .

```
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

### *Contrôle d'accès à granularité fine*

Les clients MQTT sont intégrés à un système de contrôle d'accès existant, qui autorise les groupes à effectuer des actions sur les objets.

## Pourquoi et quand exécuter cette tâche

Un ID utilisateur est affecté à un ou plusieurs groupes de système d'exploitation en fonction des autorisations requises. Si les applications IBM MQ publient et s'abonnent au même espace de sujet que les clients MQTT , utilisez ce modèle. Les groupes sont appelés Publish X, Subscribe Y et mqtt

### **Publish X**

Les membres des groupes Publish X peuvent publier dans *topicX*.

### **Subscribe Y**

Les membres des groupes Subscribe Y peuvent s'abonner à *topicY*.

## mqtt

Les membres du groupe *mqtt* peuvent envoyer des publications aux clients MQTT .

### Procédure

1. Créez plusieurs groupes, Publish *X* et Subscribe *Y*, qui sont alloués à plusieurs rubriques d'administration dans l'arborescence de rubriques de publication / abonnement.
2. Créez un groupe *mqtt*.
3. Créez plusieurs ID utilisateur, *mqttUsers*, et ajoutez les utilisateurs à n'importe quel groupe, en fonction de ce qu'ils sont autorisés à faire.
4. Autorisez différents groupes Publish *X* et Subscribe *X* à accéder à des rubriques différentes et autorisez le groupe *mqtt* à envoyer des messages aux clients MQTT .

```
setmqaut -m qMgr -t topic -n topic1 -p Publish X -all +pub
setmqaut -m qMgr -t topic -n topic1 -p Subscribe X -all +pub +sub
setmqaut -m qMgr -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p mqtt -all +put
```

### Windows Linux AIX MQTT Authentification du client à l'aide d'un mot de passe

Authentifiez le Nom d'utilisateur à l'aide du mot de passe du client. Vous pouvez authentifier le client en utilisant une identité différente de celle utilisée pour autoriser le client à diffuser des publications et à s'abonner aux sujets.

Le service de télémétrie (MQXR) utilise JAAS pour authentifier le Nom d'utilisateur du client. JAAS utilise le mot de passe fourni par le client MQTT .

L'administrateur IBM MQ décide si le Nom d'utilisateur doit être authentifié ou non, en configurant le canal MQTT auquel se connecte un client. Les clients peuvent se voir attribuer des canaux différents et chaque canal peut être configuré pour authentifier ses clients de différentes façons. A l'aide de JAAS, vous pouvez décider quelles sont les méthodes qui doivent authentifier le client, et celles qui peuvent authentifier le client.

Le choix de l'identité pour l'authentification n'affecte pas le choix de l'identité pour l'autorisation. Vous pouvez définir une identité commune pour l'autorisation à des fins d'administration, mais authentifier chaque utilisateur afin qu'il puisse utiliser cette identité. La procédure suivante met en évidence les étapes permettant d'authentifier des utilisateurs individuels pour l'utilisation d'une identité commune :

1. L'administrateur IBM MQ définit l'identité du canal MQTT sur n'importe quel nom, tel que MQTTClientUser, à l'aide de IBM MQ Explorer.
2. L'administrateur IBM MQ autorise MQTTClient à diffuser des publications et à s'abonner à n'importe quelle rubrique :

```
setmqaut -m QM1 -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p MQTTClient -all +put
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p MQTTClient -all +pub +sub
```

3. Le développeur d'applications client MQTT crée un objet MqttConnectOptions et définit Nom d'utilisateur et Mot de passe avant de se connecter au serveur.
4. Le développeur de sécurité crée un JAAS LoginModule pour authentifier le nom d'utilisateur avec le mot de passe et l'inclut dans le fichier de configuration JAAS .
5. L'administrateur IBM MQ configure le canal MQTT pour authentifier le UserName du client à l'aide de JAAS.



Les connexions entre le client MQTT et le gestionnaire de files d'attente sont toujours initiées par le client MQTT. Le client MQTT est toujours le client SSL. L'authentification de client du serveur et l'authentification de serveur du client MQTT sont toutes les deux facultatives.

En fournissant au client un certificat numérique signé privé, vous pouvez authentifier le client MQTT auprès d'IBM MQ. L'administrateur IBM MQ peut forcer les clients MQTT à s'authentifier auprès du gestionnaire de files d'attente à l'aide de TLS. Vous pouvez uniquement demander l'authentification de client comme faisant partie d'une authentification mutuelle.

Comme alternative à l'utilisation de SSL, certains types de réseaux privés virtuels (VPN), tels que IPsec, authentifient les noeuds finaux d'une connexion TCP/IP. Le réseau privé virtuel VPN chiffre chaque paquet transitant sur le réseau. Une fois qu'une telle connexion VPN est établie, vous avez établi un réseau sécurisé. Vous pouvez connecter les clients MQTT aux canaux de télémétrie avec TCP/IP sur le réseau VPN.

L'authentification de client à l'aide de TLS suppose que le client possède une clé confidentielle. La clé confidentielle est la clé privée du client dans le cas d'un certificat autosigné ou une clé fournie par une autorité de certification. La clé permet de signer le certificat numérique du client. Toute personne en possession de la clé publique correspondante peut vérifier le certificat numérique. Les certificats peuvent être accrédités, ou s'ils font partie d'une chaîne, il est possible de remonter par l'intermédiaire de la chaîne jusqu'au certificat racine accrédité. L'opération de vérification du client envoie tous les certificats de la chaîne fournie par le client au serveur. Le serveur vérifie la chaîne de certificats jusqu'à ce qu'il trouve un certificat digne de confiance. Il s'agit soit du certificat public généré par un certificat autosigné, soit d'un certificat racine généralement émis par une autorité de certification. Au cours de l'étape finale facultative le certificat de confiance peut être comparé à une liste de révocation de certificats "en direct".

Le certificat de confiance peut être émis par une autorité de certification et inclus dans l'espace de stockage de certificats de JRE. Il peut s'agir d'un certificat autosigné ou de tout certificat qui a été ajouté au magasin de clés de canal de télémétrie en tant que certificat digne de confiance.

**Remarque :** Le canal de télémétrie dispose d'un magasin servant à la fois de magasin de clés et de magasin de clés de confiance, qui contient les clés privées d'accès à un ou plusieurs canaux de télémétrie et les certificats publics nécessaires pour authentifier les clients. Comme un canal SSL doit être associé à un magasin de clés et que ce magasin est le même fichier que le magasin de clés de confiance du canal, l'espace de stockage des certificats du JRE n'est jamais référencé. Il en résulte que si l'authentification d'un client requiert un certificat racine de l'autorité de certification, vous devez placer ce certificat dans le magasin de clés du canal, même si le certificat racine de l'autorité de certification est déjà présent dans l'espace de stockage des certificats du JRE. L'espace de stockage des certificats du JRE n'est jamais référencé.

Ayez à l'esprit les menaces que l'authentification de client peut déjouer et le rôle du client et du serveur pour faire échouer ces menaces. L'authentification du certificat client en soi ne suffit pas à empêcher l'accès non autorisé à un système. Si quelqu'un d'autre a pris la possession du dispositif client, ce dernier ne s'exécute pas nécessairement avec les droits du détenteur du certificat. Ne faites jamais confiance à une protection unique contre des attaques non désirées. Utilisez au moins une approche de type authentification à deux facteurs et ajoutez en supplément à la possession d'un certificat la connaissance d'informations privées. Par exemple, utilisez JAAS, et authentifiez le client à l'aide d'un mot de passe émis par le serveur.

La principale menace à laquelle est confronté le certificat client est de se trouver aux mains de quelqu'un de mal intentionné. Le certificat se trouve dans un magasin de clés protégé par mot de passe au niveau du client. Comment se retrouve-t-il dans le magasin de clés ? Comment le client MQTT parvient-il à se procurer le mot de passe du magasin de clés ? Quel est le degré de sécurité de la protection par mot de passe ? Les dispositifs de télémétrie sont faciles à supprimer et peuvent donc être piratés en privé. Le dispositif matériel peut-il être imperméable aux actes de malveillance ? La distribution et la protection de certificat côté client sont reconnues comme un acte difficile, on parle alors de problème gestion de clés.

La menace secondaire est le cas où le dispositif est mal utilisé pour accéder au serveur de manière non intentionnelle. Par exemple, si l'application MQTT est trafiquée il est possible d'utiliser une faiblesse dans la configuration du serveur à l'aide de l'identité du client authentifié.

Pour authentifier un client MQTT à l'aide de SSL, configurez le canal de télémétrie et le client.

### Concepts associés

[«Configuration du canal de télémétrie pour l'authentification du client MQTT à l'aide de TLS», à la page 362](#)

L'administrateur IBM MQ configure les canaux de télémétrie sur le serveur. Chaque canal est configuré de manière à accepter une connexion TCP/IP sur un numéro de port différent. Les canaux TLS sont configurés avec accès au fichier de clés protégé par phrase passe. Si un canal TLS est défini sans phrase passe ou fichier de clés, le canal n'accepte pas de connexion TLS.

[Configuration du client MQTT pour l'authentification du client à l'aide de TLS](#)

### **Configuration du canal de télémétrie pour l'authentification du client MQTT à l'aide de TLS**

L'administrateur IBM MQ configure les canaux de télémétrie sur le serveur. Chaque canal est configuré de manière à accepter une connexion TCP/IP sur un numéro de port différent. Les canaux TLS sont configurés avec accès au fichier de clés protégé par phrase passe. Si un canal TLS est défini sans phrase passe ou fichier de clés, le canal n'accepte pas de connexion TLS.

Définissez la propriété `com.ibm.mq.MQTT.ClientAuth` d'un canal de télémétrie TLS sur `REQUIRED` pour forcer tous les clients à se connecter sur ce canal afin de fournir la preuve qu'ils ont vérifié les certificats numériques. Les certificats client sont authentifiés à l'aide des certificats des autorités de certification, ce qui permet d'obtenir un certificat racine digne de confiance. Si le certificat client est autosigné ou signé par un certificat provenant d'une autorité de certification, les certificats signés publiquement du client ou de l'autorité de certification doivent être stockés de manière sécurisée sur le serveur.

Placez le certificat client signé publiquement ou le certificat de l'autorité de certification dans le magasin de clés du canal de télémétrie. Sur le serveur, les certificats signés publiquement sont stockés dans le même fichier de clés que les certificats signés en privé, plutôt que dans un magasin de clés de confiance distinct.

Le serveur vérifie la signature des certificats client qu'il envoie à l'aide de tous les certificats publics et des suites de chiffrement dont il dispose. Le serveur vérifie la chaîne de clés. Le gestionnaire de files d'attente peut être configuré pour tester le certificat par rapport à la liste de révocation de certificat. La propriété de liste de noms de révocation du gestionnaire de files d'attente est `SSLCRLNL`.

Si l'un des certificats envoyés par un client est vérifié par un certificat dans le magasin de clés du serveur, le client est authentifié.

L'administrateur IBM MQ peut configurer le même canal de télémétrie pour utiliser `JAAS` afin de vérifier le `UserName` ou `ClientIdentifier` du client avec le mot de passe du client.

Vous pouvez utiliser le même magasin de clés pour plusieurs canaux de télémétrie.

La vérification d'au moins un certificat numérique dans le magasin de clés du client protégé par mot de passe sur le périphérique authentifie le client auprès du serveur. Le certificat numérique est utilisé uniquement pour l'authentification par IBM MQ. Il n'est pas utilisé pour vérifier l'adresse TCP/IP du client, ni pour définir l'identité du client pour l'autorisation ou la comptabilité. L'identité du client adoptée par le serveur est `Username` ou `ClientIdentifier` du client, ou une identité créée par l'administrateur IBM MQ.

Vous pouvez également utiliser des suites de chiffrement TLS pour l'authentification du client. Si vous envisagez d'utiliser des suites de chiffrement SHA-2, voir [«Configuration système requise pour utiliser les suites de chiffrement SHA-2 avec les canaux MQTT», à la page 367](#).

### Concepts associés

[«Configuration de canal de télémétrie pour l'authentification de canal à l'aide de TLS», à la page 363](#)

L'administrateur IBM MQ configure les canaux de télémétrie sur le serveur. Chaque canal est configuré de manière à accepter une connexion TCP/IP sur un numéro de port différent. Les canaux TLS sont configurés avec accès au fichier de clés protégé par phrase passe. Si un canal TLS est défini sans phrase passe ou fichier de clés, le canal n'accepte pas de connexion TLS.

[CipherSpecs et CipherSuites](#)

### Référence associée

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

## Windows Linux AIX **Authentification de canal de télémétrie à l'aide de TLS**

Les connexions entre le client MQTT et le gestionnaire de files d'attente sont toujours initiées par le client MQTT. Le client MQTT est toujours le client SSL. L'authentification de client du serveur et l'authentification de serveur du client MQTT sont toutes les deux facultatives.

Le client tente toujours d'authentifier le serveur, sauf si le client est configuré pour utiliser un CipherSpec qui prend en charge la connexion anonyme. Si l'authentification échoue, la connexion n'est pas établie.

Comme alternative à l'utilisation de SSL, certains types de réseaux privés virtuels (VPN), tels que IPsec, authentifient les noeuds finaux d'une connexion TCP/IP. Le réseau privé virtuel VPN chiffre chaque paquet transitant sur le réseau. Une fois qu'une telle connexion VPN est établie, vous avez établi un réseau sécurisé. Vous pouvez connecter les clients MQTT aux canaux de télémétrie avec TCP/IP sur le réseau VPN.

L'authentification de serveur à l'aide de SSL permet d'authentifier le serveur auquel vous êtes sur le point d'envoyer des informations confidentielles. Le client effectue les vérifications correspondant aux certificats envoyés à partir du serveur, par rapport aux certificats placés dans son magasin de clés de confiance ou dans son magasin JRE cacerts .

Le magasin de certificats JRE est un fichier JKS, cacerts. Il se trouve dans JRE InstallPath\lib\security\ . Il est installé avec le mot de passe par défaut changeit. Vous pouvez stocker les certificats que vous accédez dans l'espace de stockage des certificats JRE ou dans le magasin de clés de confiance. Vous ne pouvez pas utiliser les deux magasins. Utilisez le magasin de clés de confiance du client si vous souhaitez conserver les certificats publics que le client accède séparément des certificats utilisés par d'autres applications Java . Utilisez le magasin de certificats JRE si vous souhaitez utiliser un magasin de certificats commun pour toutes les applications Java exécutées sur le client. Si vous décidez d'utiliser cet espace de stockage, vérifiez les certificats qu'il contient afin d'être sûr que vous les accédez.

Vous pouvez modifier la configuration JSSE en indiquant un autre fournisseur d'accréditation. Vous pouvez personnaliser un fournisseur d'accréditation pour effectuer différentes vérifications sur un certificat. Dans certains environnements OSGi qui ont utilisé le client MQTT, l'environnement fournit un autre fournisseur de relations de confiance.

Pour authentifier un canal de télémétrie à l'aide de TLS, configurez le serveur et le client.



## Windows Linux AIX **Configuration de canal de télémétrie pour l'authentification de canal à l'aide de TLS**

L'administrateur IBM MQ configure les canaux de télémétrie sur le serveur. Chaque canal est configuré de manière à accepter une connexion TCP/IP sur un numéro de port différent. Les canaux TLS sont configurés avec accès au fichier de clés protégé par phrase passe. Si un canal TLS est défini sans phrase passe ou fichier de clés, le canal n'accepte pas de connexion TLS.

Stockez le certificat numérique du serveur, signé avec sa clé privée, dans le magasin de clés que le canal de télémétrie va utiliser sur le serveur. Stockez tous les certificats de sa chaîne de clés dans le magasin de clés, si vous souhaitez transmettre la chaîne de clés au client. Configurez le canal de télémétrie à l'aide de l'explorateur IBM MQ pour utiliser TLS. Indiquez le chemin d'accès au magasin de clés et la phrase passe

pour accéder au magasin de clés. Si vous ne définissez pas le numéro de port TCP/IP du canal, le numéro de port du canal de télémétrie TLS est par défaut 8883.

Vous pouvez également utiliser des suites de chiffrement TLS pour l'authentification de canal. Si vous envisagez d'utiliser des suites de chiffrement SHA-2, voir «[Configuration système requise pour utiliser les suites de chiffrement SHA-2 avec les canaux MQTT](#)», à la page 367.

**Important :**   Depuis la IBM MQ 9.4.0, les référentiels de clés et les fichiers de dissimulation CMS ne sont pas pris en charge avec les canaux AMQP et MQTT qui utilisent SSL/TLS. Utilisez les référentiels de clés PKCS #12 et protégez les mots de passe des référentiels de clés à l'aide du système de protection par mot de passe IBM MQ . Vous pouvez créer un référentiel de clés PKCS #12 à l'aide de la commande suivante:

```
runmqakm -keydb -create -db filename.p12 -pw password -type pkcs12
```

Cette commande crée un fichier de référentiel de clés PKCS #12 nommé *filename.p12* qui est sécurisé avec le mot de passe spécifié.

### Concepts associés

«[Configuration du canal de télémétrie pour l'authentification du client MQTT à l'aide de TLS](#)», à la page 362

L'administrateur IBM MQ configure les canaux de télémétrie sur le serveur. Chaque canal est configuré de manière à accepter une connexion TCP/IP sur un numéro de port différent. Les canaux TLS sont configurés avec accès au fichier de clés protégé par phrase passe. Si un canal TLS est défini sans phrase passe ou fichier de clés, le canal n'accepte pas de connexion TLS.

[CipherSpecs et CipherSuites](#)

### Référence associée

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)



## Exemple de configuration de canal MQTT à l'aide de l'authentification TLS

Cet exemple vous guide dans la configuration d'un canal MQTT qui utilise l'authentification TLS.

L'exemple configure un canal entre MQTT et Mosquitto.

L'exemple utilise un conteneur Docker pour IBM MQ sur Red Hat Enterprise Linux et Mosquitto sur CentOS, mais s'applique à tout type de serveur. (CentOS a été utilisé pour Mosquitto en raison des droits d'accès au registre.)

## Configuration du magasin de clés et du canal IBM MQ pour TLS unidirectionnel

**Important :**   Depuis la IBM MQ 9.4.0, les référentiels de clés et les fichiers de dissimulation CMS ne sont pas pris en charge avec les canaux AMQP et MQTT qui utilisent SSL/TLS. Utilisez les référentiels de clés PKCS #12 et protégez les mots de passe des référentiels de clés à l'aide du système de protection par mot de passe IBM MQ .

Procédez comme suit :

1.   Créez le magasin de clés IBM MQ :



```
runmqakm -keydb -create -db mqtt.p12 -pw "passw0rd" -type p12
```

2.   Créez un certificat personnel:

```
runmqakm -cert -create -db mqtt.p12 -pw "passw0rd" -size 2048 -dn "CN= mqm, OU=MQTest, O=MQSupport, C=US" -sig_alg SHA256_WITH_RSA -label ibmwebspheremqmqm
```

Vous pouvez utiliser la commande suivante pour confirmer la création du certificat:

```
runmqakm -cert -list -v -db mqtt.p12 -pw "passw0rd"
```

3.   Créez le canal MQTT en entrant la commande suivante à l'invite runmqsc:

```
DEFINE CHANNEL(MQTTDEMO) CHLTYPE(MQTT) BACKLOG(4096) PORT(8883) MCAUSER('mqm')
PROTOCOL(MQTTV311,MQTTV3,HTTP) SSLCAUTH(OPTIONAL) SSLCIPH('SSL_RSA_WITH_AES_256_CBC_SHA256')
SSLKEYR('/var/mqm/mqtt/mqtt.p12') SSLKEYP('passwd') TRPTYPE(TCP)
```

Notez que le canal utilise des mappages de chiffrement Java . Voir [TLS CipherSpecs et CipherSuites](#) dans IBM MQ classes for JMS.

4. Extrayez le certificat:

```
runmqakm -cert -extract -db mqtt.kdb -stashed -label ibmwebspheremqmqm -target serverCert.pem
```

## Installez Mosquito sur CentOS dans un conteneur Docker

Pour créer un conteneur Docker avec Mosquitto exécuté sur CentOS, procédez comme suit:

1. `docker pull centos`
2. `docker run -it centos /bin/bash`
3. `yum -y install epel-release`
4. `yum -y install mosquitto`

## Déplacer le certificat de signataire vers Mosquitto

Procédez comme suit pour déplacer le certificat que vous avez créé dans IBM MQ vers Mosquitto. Ces étapes sont exécutées sur la machine hôte Docker .

1. Affichez les ID de conteneur sur Docker:

```
docker container ls
```

2. Copiez le fichier depuis le conteneur Docker vers votre système local docker

```
cp MQ_Container_ID:/var/mqm/mqtt/serverCert.pem serverCert.pem
```

3. Copiez le fichier depuis votre machine locale dans le répertoire racine de la machine centOS :

```
docker cp serverCert.pem CentOS_ContainerID:/serverCert.pem
```

## Publier avec Mosquitto

Publiez un message de test sur Mosquitto à l'aide de la commande suivante:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t
test -m 'test message' -d
```

Les arguments de commande ont la signification suivante:

**-h**

Adresse IP de l'hôte Red Hat Enterprise Linux (disponible à l'aide de **nslookup**).

**-- cafile**

Fichier contenant le certificat de signataire.

**-- non sécurisé**

Cette option est spécifiée car l'exemple utilise un certificat autosigné. N'utilisez pas cette option lorsque vous utilisez des certificats d'autorité de certification réels.

**-p**

Numéro de port.

**-i**

ID client.

- t**  
Rubrique dans laquelle la publication est en cours.
- m**  
Message en cours de publication.
- d**  
Activez les messages de débogage.

## Configuration du canal MQTT pour l'authentification TLS mutuelle

Entrez la commande suivante pour reconfigurer le canal MQTT en tant que SSLCAUTH (REQUIRED).

```
ALTER CHANNEL(MQTTDEMO) CHLTYPE(MQTT) SSLCAUTH(REQUIRED)
```

## Créez une paire clé / certificat sur le serveur Mosquitto et ajoutez-la à IBM MQ

Entrez les commandes suivantes pour créer la paire clé / certificat sur Mosquitto:

1. Utilisez **openssl** pour créer la paire clé / certificat pour Mosquitto:

```
openssl req -x509 -newkey rsa:4096 -keyout mosquittoKey.pem -out mosquittoCert.pem -subj "/CN=Mosquitto"
```

2. Répertoriez les ID de conteneur pour les conteneurs:

```
docker container ls
```

3. Copiez le certificat Mosquitto dans votre docker de système local:

```
docker cp CentOS_ContainerID:mosquittoCert.pem .
```

4. Copiez le certificat Mosquitto dans IBM MQ:

```
docker cp mosquittoCert.pem MQ_Container_ID:/var/mqm/mqtt
```

5. Ajoutez le certificat au magasin de clés IBM MQ :

```
runmqakm -cert -add -db mqtt.kdb -stashed -file mosquittoCert.pem
```

6. Redémarrez le canal MQTT .

## Publier avec Mosquitto et authentification mutuelle

Procédez comme suit pour publier avec Mosquitto à l'aide de l'authentification mutuelle.

1. La commande suivante doit publier un message de test:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d --cert mosquittoCert.pem --key mosquittoKey.pem
```

2. La commande suivante ne doit pas publier de message de test et générer un message d'erreur car elle n'envoie pas de certificat personnel de Mosquitto:

```
mosquitto_pub -h 172.17.0.2 --cafile serverCert.pem --insecure -p 8883 -i mosquittoClient -t test -m 'test message' -d /var/mqm/qmgrs/mqttDemoQM/errors/ mqxr_0.log
```

### Information associée

[Gestion des clés et des certificats](#)

## Configuration système requise pour utiliser les suites de chiffrement SHA-2 avec les canaux MQTT

Si vous utilisez une version de Java qui prend en charge les suites de chiffrement SHA-2, vous pouvez utiliser ces suites pour sécuriser vos canaux MQTT (téléométrie) et vos applications client.

Pour IBM MQ 8.0, qui inclut le service de téléométrie (MQXR), la version minimale de Java est Java 7 à partir de IBM, SR6. Les suites de chiffrement SHA-2 sont prises en charge par défaut dans Java 7 à partir de IBM, SR4. Vous pouvez donc utiliser des suites de chiffrement SHA-2 avec le service de téléométrie (MQXR) pour sécuriser vos canaux MQTT (téléométrie).

Si vous exécutez un client MQTT avec un JRE différent, vous devez vous assurer qu'il prend également en charge les suites de chiffrement SHA-2.

### Concepts associés

[Service de téléométrie \(MQXR\)](#)

«Configuration de canal de téléométrie pour l'authentification de canal à l'aide de TLS», à la page 363  
L'administrateur IBM MQ configure les canaux de téléométrie sur le serveur. Chaque canal est configuré de manière à accepter une connexion TCP/IP sur un numéro de port différent. Les canaux TLS sont configurés avec accès au fichier de clés protégé par phrase passe. Si un canal TLS est défini sans phrase passe ou fichier de clés, le canal n'accepte pas de connexion TLS.

### Référence associée

[DEFINE CHANNEL \(MQTT\)](#)

[ALTER CHANNEL \(MQTT\)](#)

## Confidentialité de la publication sur les canaux de téléométrie

La confidentialité des publications MQTT envoyées dans les deux sens sur les canaux de téléométrie est sécurisée à l'aide de TLS pour chiffrer les transmissions sur la connexion.

Les clients MQTT qui se connectent aux canaux de téléométrie utilisent TLS pour sécuriser la confidentialité des publications transmises sur le canal à l'aide de la cryptographie à clé symétrique. Etant donné que les noeuds finaux ne sont pas authentifiés, vous ne pouvez pas accréditer le chiffrement de canal seul. Combinez la sécurisation de la confidentialité avec l'authentification du serveur ou l'authentification mutuelle.

Comme alternative à l'utilisation de SSL, certains types de réseaux privés virtuels (VPN), tels que IPsec, authentifient les noeuds finaux d'une connexion TCP/IP. Le réseau privé virtuel VPN chiffre chaque paquet transitant sur le réseau. Une fois qu'une telle connexion VPN est établie, vous avez établi un réseau sécurisé. Vous pouvez connecter les clients MQTT aux canaux de téléométrie avec TCP/IP sur le réseau VPN.

Pour une configuration classique, qui chiffre le canal et authentifie le serveur, consultez la section «Authentification de canal de téléométrie à l'aide de TLS», à la page 363.

Le chiffrement des connexions TLS sans authentification du serveur expose la connexion aux attaques man-in-the-middle. Bien que les informations que vous échangez soient protégées contre les écoutes clandestines, vous ne savez pas avec qui vous réalisez les échanges. A moins que vous ne contrôliez le réseau, vous êtes exposé à ce que quelqu'un intercepte vos transmissions IP, en se faisant passer pour un noeud final.

Vous pouvez créer une connexion TLS chiffrée, sans authentifier le serveur, à l'aide d'un CipherSpec d'échange de clés Diffie-Hellman qui prend en charge le protocole TLS anonyme. Le secret maître, partagé entre le client et le serveur et utilisé pour chiffrer les transmissions TLS, est établi sans échanger de certificat de serveur signé de manière privée.

Etant donné que les connexions anonymes ne sont pas sécurisées, la plupart des implémentations TLS n'utilisent pas par défaut les CipherSpecs anonymes. Si une demande client de connexion TLS est acceptée par un canal de téléométrie, le canal doit disposer d'un magasin de clés protégé par une phrase

passé. Par défaut, les implémentations TLS n'utilisant pas de CipherSpecs anonyme, le magasin de clés doit contenir un certificat signé de manière privée que le client peut authentifier.

Si vous utilisez des spécifications CipherSpecs anonymes, le magasin de clés du serveur doit exister, mais il ne doit pas forcément contenir des certificats signés de manière privée.

Une autre façon d'établir une connexion cryptée consiste à remplacer le fournisseur d'accréditation côté client par votre propre implémentation. Votre fournisseur d'accréditation n'effectuerait pas l'authentification du certificat serveur, mais la connexion serait cryptée.



**Avertissement :** Lorsque vous utilisez TLS avec MQTT, vous pouvez utiliser des messages volumineux, mais cela peut avoir un impact possible sur les performances. MQTT est optimisé pour le traitement des messages de petite taille (généralement entre 1KB et 1MB).

Windows

Linux

AIX

## Configuration TLS des clients MQTT Java et des canaux de télémétrie

Configurez TLS pour authentifier le canal de télémétrie et le client MQTT Java, et chiffrez le transfert des messages entre eux. MQTT Les clients Java utilisent JSSE (Java Secure Socket Extension) pour connecter les canaux de télémétrie à l'aide de TLS. Comme alternative à l'utilisation de SSL, certains types de réseaux privés virtuels (VPN), tels que IPsec, authentifient les noeuds finaux d'une connexion TCP/IP. Le réseau privé virtuel VPN chiffre chaque paquet transitant sur le réseau. Une fois qu'une telle connexion VPN est établie, vous avez établi un réseau sécurisé. Vous pouvez connecter les clients MQTT aux canaux de télémétrie avec TCP/IP sur le réseau VPN.

Vous pouvez configurer la connexion entre un client Java MQTT et un canal de télémétrie pour utiliser le protocole TLS sur TCP/IP. Ce qui est sécurisé dépend de la façon dont vous configurez TLS pour utiliser JSSE. En partant de la configuration la plus sécurisée, vous pouvez configurer trois niveaux différents de sécurité :

1. Autorisez uniquement les clients MQTT dignes de confiance à se connecter. Connectez un client MQTT uniquement à un canal de télémétrie sécurisé. Chiffrez les messages entre le client et le gestionnaire de files d'attente ; voir [«Authentification du client MQTT à l'aide de TLS»](#), à la page 361
2. Connectez un client MQTT uniquement à un canal de télémétrie sécurisé. Chiffrez les messages entre le client et le gestionnaire de files d'attente ; voir [«Authentification de canal de télémétrie à l'aide de TLS»](#), à la page 363.
3. Chiffrez les messages entre le client et le gestionnaire de files d'attente ; voir [«Confidentialité de la publication sur les canaux de télémétrie»](#), à la page 367.

### Paramètres de configuration JSSE

Modifiez les paramètres JSSE pour modifier la manière dont une connexion TLS est configurée. Les paramètres de configuration JSSE sont composés de trois ensembles :

1. [Canal MQ Telemetry](#)
2. [clientMQTT Java](#)
3. [JRE](#)

Configurez les paramètres du canal de télémétrie à l'aide d'IBM MQ Explorer. Définissez les paramètres du client MQTT Java dans l'attribut `MqttConnectionOptions.SSLProperties`. Modifiez les paramètres de sécurité en éditant les fichiers dans le répertoire de sécurité de JRE à la fois sur le client et sur le serveur.

### MQ Telemetry canal

Définissez tous les paramètres TLS du canal de télémétrie à l'aide de IBM MQ Explorer.

#### ChannelName

ChannelName est un paramètre obligatoire sur tous les canaux.



Le nom de canal identifie le canal associé à un numéro de port particulier. Nommez des canaux pour vous aider à administrer des ensembles de clients MQTT .

### **PortNumber**

PortNumber est un paramètre facultatif sur tous les canaux. La valeur par défaut est 1883 pour les canaux TCP et 8883 pour les canaux TLS.

Numéro de port TCP/IP associé à ce canal. Les clients MQTT sont connectés à un canal en spécifiant le port défini pour le canal. Si le canal possède des propriétés TLS, le client doit se connecter à l'aide du protocole TLS ; par exemple:

```
MQTTClient mqttClient = new MqttClient( "ssl://www.example.org:8884", "clientId1");
mqttClient.connect();
```

### **KeyFileName**

KeyFileName est un paramètre obligatoire pour les canaux TLS. Il doit être omis pour les canaux TCP.

KeyFileName est le chemin d'accès au magasin de clés Java contenant les certificats numériques que vous fournissez. Sur le serveur, utilisez les types de magasin de clés JKS, JCEKS ou PKCS12.

Identifiez le type de magasin de clés à l'aide de l'une des extensions de fichier suivantes :

- .jks
- .jceks
- .p12
- .pkcs12

Un magasin de clés associé à une autre extension est censé être un magasin JKS.

Vous pouvez combiner un magasin de clés d'un type donné se trouvant sur le serveur avec d'autres types de magasins de clés se trouvant sur le client.

Placez le certificat privé du serveur dans le magasin de clés. Le certificat est connu sous le nom de certificat serveur. Ce certificat peut être autosigné ou faire partie d'une chaîne de certificats signée par une autorité de signature.

Si vous utilisez une chaîne de certificats, placez les certificats associés dans le magasin de clés du serveur.

Le certificat du serveur et tous les certificats de la chaîne de certificats sont envoyés aux clients afin d'authentifier l'identité du serveur.

Si vous avez attribué à ClientAuth la valeur Required, le magasin de clés doit contenir tous les certificats nécessaires à l'authentification du client. Le client envoie un certificat autosigné, ou une chaîne de certificats, et est authentifié par la première vérification de cet élément par rapport à un certificat se trouvant dans le magasin de clés. Lorsque vous utilisez une chaîne de certificats, un certificat peut vérifier plusieurs clients, même s'ils sont émis avec des certificats client différents.

### **PassPhrase**

PassPhrase est un paramètre obligatoire pour les canaux TLS. Il doit être omis pour les canaux TCP.

La phrase passe est utilisée pour protéger le magasin de clés.

### **ClientAuth**

ClientAuth est un paramètre TLS facultatif. Par défaut, aucune authentification de client n'est effectuée. Il doit être omis pour les canaux TCP.

Définissez ClientAuth si vous voulez que le service de télémétrie (MQXR) authentifie le client avant de lui permettre de se connecter au canal de télémétrie.

Si vous définissez `ClientAuth`, le client doit se connecter au serveur à l'aide de TLS et authentifier le serveur. En réponse à la définition de `ClientAuth`, le client envoie son certificat numérique au serveur et tout autre certificat dans son magasin de clés. Son certificat numérique est connu sous le nom de certificat client. Ces certificats sont authentifiés par rapport aux certificats se trouvant dans le magasin de clés du canal et dans le magasin `cacerts`.

### **CipherSuite**

`CipherSuite` est un paramètre TLS facultatif. Par défaut, il essaie toutes les spécifications `CipherSpecs` activées. Il doit être omis pour les canaux TCP.

Si vous souhaitez utiliser un `CipherSpec` particulier, définissez `CipherSuite` sur le nom du `CipherSpec` qui doit être utilisé pour établir la connexion TLS.

Le service de télémétrie et le client MQTT négocient un `CipherSpec` commun à partir de tous les `CipherSpecs` activés à chaque extrémité. Si un `CipherSpec` spécifique est indiqué à l'une ou aux deux extrémités, il doit correspondre au `CipherSpec` de l'autre extrémité.

Installez les chiffrements supplémentaires en ajoutant des fournisseurs supplémentaires à JSSE.

### **FIPS (Federal Information Processing Standards)**

FIPS est un paramètre facultatif. Par défaut, il n'est pas défini.

Utilisez le panneau de propriété du gestionnaire de files d'attente ou `runmqsc`, pour définir `SSLFIPS`. `SSLFIPS` Indique si seuls les algorithmes certifiés pour FIPS doivent être utilisés.

### **Revocation namelist**

`Revocation namelist` est un paramètre facultatif. Par défaut, il n'est pas défini.

Utilisez le panneau de propriété du gestionnaire de files d'attente ou `runmqsc`, pour définir `SSLCRLNL`. `SSLCRLNL` indique une liste de noms d'objets d'informations d'authentification utilisés pour fournir des emplacements de révocation de certificat.

Aucun autre paramètre de gestionnaire de files d'attente qui définit les propriétés TLS n'est utilisé.

## **Client MQTT Java**

Définissez les propriétés TLS du client Java dans `MqttConnectionOptions.SSLProperties`; par exemple:

```
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.keyStoreType", "JKS");
com.ibm.micro.client.mqttv3.MqttConnectOptions conOptions = new MqttConnectOptions();
conOptions.setSSLProperties(sslClientProperties);
```

Les noms et les valeurs de propriétés spécifiques sont décrits dans la classe `MqttConnectOptions`. Pour des liens vers la documentation de l'API client pour les bibliothèques client MQTT, voir [MQTT client programming reference](#).

### **Protocole**

`Protocol` est facultatif.

Le protocole est sélectionné en négociation avec le serveur de télémétrie. Si vous avez besoin d'un protocole spécifique, vous pouvez en sélectionner un. Si le serveur de télémétrie ne prend pas en charge le protocole, la connexion échoue.

### **ContextProvider**

`ContextProvider` est facultatif.

### **KeyStore**

`KeyStore` est facultatif. Configurez-le si `ClientAuth` est défini au niveau du serveur pour forcer l'authentification du client.

Placez le certificat numérique du client, signé à l'aide de la clé privée, dans le magasin de clés. Spécifiez le chemin d'accès et le mot de passe du magasin de clés. Le type et le fournisseur sont facultatifs. JKS est le type par défaut et IBMJCE est le fournisseur par défaut.

Indiquez un fournisseur de magasin de clés pour référencer une classe qui ajoute un nouveau fournisseur de magasin de clés. Transmettez le nom de l'algorithme utilisé par le fournisseur de magasin de clés pour instancier `KeyManagerFactory` en définissant le nom du gestionnaire de clés.

## TrustStore

TrustStore est facultatif. Vous pouvez placer tous les certificats que vous accédez dans le magasin `cacerts` de JRE.

Configurez le magasin de clés de confiance si vous souhaitez disposer d'un fichier de ce type distinct pour le client. Il n'est peut-être pas nécessaire de configurer le magasin de clés de confiance si le serveur utilise un certificat émis par une autorité de certification bien connue dont le certificat racine est déjà stocké dans `cacerts`.

Ajoutez le certificat signé publiquement du serveur ou du certificat racine au magasin de clés de confiance, et indiquez le chemin d'accès et le mot de passe du magasin de clés de confiance. JKS est le type par défaut et IBMJCE est le fournisseur par défaut.

Indiquez un fournisseur de magasin de clés de confiance pour référencer une classe qui ajoute un nouveau fournisseur de magasin de clés de confiance. Transmettez le nom de l'algorithme utilisé par le fournisseur de magasin de clés de confiance pour instancier `TrustManagerFactory` en définissant le nom du gestionnaire d'accréditation.



## JRE



D'autres aspects de la sécurité Java qui affectent le comportement de TLS sur le client et le serveur sont configurés dans l'environnement d'exécution Java. Les fichiers de configuration sous Windows se trouvent dans `Java Installation Directory\jre\lib\security`. Si vous utilisez le JRE livré avec IBM MQ, le chemin est celui qui est indiqué dans le tableau suivant :

Plateforme	Chemin d'accès au fichier
Windows	<code>WMQ Installation Directory\java\jre\lib\security</code>
AIX and Linux plateformes	<code>WMQ Installation Directory/java/jre64/jre/lib/security</code>

## Autorités de certification reconnues

Le fichier `cacerts` contient les certificats racine des autorités de certification reconnues. Le fichier `cacerts` est utilisé par défaut, sauf si vous définissez un autre magasin de clés de confiance. Si vous utilisez le fichier `cacerts` ou que vous n'indiquez aucun magasin de clés de confiance, vous devez vérifier et éditer la liste des signataires du fichier `cacerts` pour remplir les conditions requises en matière de sécurité.

 `V 9.4.0`  `V 9.4.0` Vous pouvez utiliser la commande `runmqktool` pour gérer le fichier de certificats `cacerts`. `cacerts` est un fichier JKS. Spécifiez le paramètre `-storetype jks` lorsque la commande `runmqktool` est utilisée pour gérer le fichier de certificats.

 `V 9.4.0`  `V 9.4.0` Le mot de passe par défaut du fichier `cacerts` est `changeit`. Modifiez le mot de passe à l'aide de la commande `runmqktool -storepasswd` pour sécuriser le fichier.

## Configuration des classes de sécurité

Utilisez le fichier `java.security` pour enregistrer les fournisseurs de sécurité supplémentaires et d'autres propriétés de sécurité par défaut.

### Droits d'accès

Utilisez le fichier `java.policy` pour modifier les droits d'accès accordés aux ressources. `javaws.policy` accorde des droits à `javaws.jar`

### Puissance de chiffrement

Certains JRE sont livrés avec une puissance de chiffrement réduite. Si vous ne pouvez pas importer les clés dans les magasins de clés, la réduction de la puissance de chiffrement peut en être la cause. Si nécessaire, téléchargez des fichiers de juridiction forte mais limitée à partir des [kits de développement IBM](#), informations de sécurité.

**Important :** Votre pays d'origine est peut-être assujéti à des restrictions relatives à l'importation, la possession, l'utilisation ou la réexportation de logiciel de chiffrement vers un autre pas. Avant de télécharger ou d'utiliser des fichiers de règles sans restriction, vous devez vérifier les lois applicables dans votre pays. Vérifiez ses règlements et sa politique en matière d'importation, de possession, d'utilisation et de réexportation de logiciel de chiffrement afin de déterminer si cette action est autorisée.

## Modifiez le fournisseur d'accréditation pour permettre au client de se connecter à n'importe quel serveur.

L'exemple montre comment ajouter un fournisseur d'accréditation et le référencer à partir du code client MQTT. Dans l'exemple aucune authentification du client ou du serveur n'est effectuée. La connexion TLS résultante est chiffrée sans être authentifiée.

Le fragment de code dans [Figure 16](#), à la [page 372](#) définit le fournisseur de relations de confiance et le gestionnaire de relations de confiance `AcceptAllProviders` pour le client MQTT.

```
java.security.Security.addProvider(new AcceptAllProvider());
java.util.Properties sslClientProperties = new Properties();
sslClientProperties.setProperty("com.ibm.ssl.trustManager", "TrustAllCertificates");
sslClientProperties.setProperty("com.ibm.ssl.trustStoreProvider", "AcceptAllProvider");
conOptions.setSSLProperties(sslClientProperties);
```

Figure 16. Fragment de code du client MQTT

```
package com.ibm.mq.id;
public class AcceptAllProvider extends java.security.Provider {
private static final long serialVersionUID = 1L;
public AcceptAllProvider() {
super("AcceptAllProvider", 1.0, "Trust all X509 certificates");
put("TrustManagerFactory.TrustAllCertificates",
AcceptAllTrustManagerFactory.class.getName());
}
}
```

Figure 17. `AcceptAllProvider.java`

```
protected static class AcceptAllTrustManagerFactory extends
javax.net.ssl.TrustManagerFactorySpi {
public AcceptAllTrustManagerFactory() {}
protected void engineInit(java.security.KeyStore keystore) {}
protected void engineInit(
javax.net.ssl.ManagerFactoryParameters parameters) {}
protected javax.net.ssl.TrustManager[] engineGetTrustManagers() {
return new javax.net.ssl.TrustManager[] { new AcceptAllX509TrustManager() };
}
}
```

Figure 18. `AcceptAllTrustManagerFactory.java`

```

protected static class AcceptAllX509TrustManager implements
javax.net.ssl.X509TrustManager {
public void checkClientTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Client authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public void checkServerTrusted(
java.security.cert.X509Certificate[] certificateChain,
String authType) throws java.security.cert.CertificateException {
report("Server authtype=" + authType);
for (java.security.cert.X509Certificate certificate : certificateChain) {
report("Accepting:" + certificate);
}
}
public java.security.cert.X509Certificate[] getAcceptedIssuers() {
return new java.security.cert.X509Certificate[0];
}
private static void report(String string) {
System.out.println(string);
}
}

```

Figure 19. AcceptAllX509TrustManager.java

Windows

Linux

AIX

## Configuration JAAS du canal de télémétrie

Configurez JAAS pour authentifier le Nom d'utilisateur envoyé par le client.

L'administrateur IBM MQ configure les canaux MQTT qui requièrent l'authentification de client à l'aide de JAAS. Indiquez le nom d'une configuration JAAS pour chaque canal qui doit effectuer une authentification JAAS. Les canaux peuvent tous utiliser la même configuration JAAS configuration, ou utiliser des configurations JAAS différentes. Les configurations sont définies dans *WMQData directory\qmgrs\qMgrName\mqxr\jaas.config*.

Le fichier *jaas.config* est organisé par nom de configuration JAAS. Sous chaque nom de configuration figure une liste de configurations de connexion ; voir «[Exemple de fichier jaas.config](#)», à la page 374.

JAAS fournit quatre normes de modules de connexion. Les modules standard NT et UNIX Login sont d'une valeur limitée.

### JndiLoginModule

S'authentifie auprès d'un service d'annuaire configuré sous JNDI (Java Naming and Directory Interface).

### Krb5LoginModule

Effectue l'authentification à l'aide des protocoles Kerberos.

### NTLoginModule

Effectue l'authentification à l'aide des informations de sécurité NT pour l'utilisateur en cours.

### UnixLoginModule

S'authentifie à l'aide des informations de sécurité UNIX pour l'utilisateur en cours.

Le problème lié à l'utilisation de *NTLoginModule* ou de *UnixLoginModule* est que le service de télémétrie (MQXR) s'exécute avec l'identité *mqm* et non avec l'identité du canal MQTT. *mqm* est l'identité transmise à *NTLoginModule* ou *UnixLoginModule* pour l'authentification et non pas l'identité du client.

Pour résoudre ce problème, écrivez votre propre module de connexion ou utilisez d'autres modules de connexion standard. Un exemple de fichier *JAASLoginModule.java* est fourni avec MQ Telemetry. Il s'agit d'une implémentation de l'interface *javax.security.auth.spi.LoginModule*. Utilisez-le pour développer votre propre méthode d'authentification.

Toute nouvelle classe *LoginModule* que vous fournissez doit figurer dans le chemin d'accès aux classes du service de télémétrie (MQXR). Ne placez pas vos classes dans les répertoires IBM MQ qui se trouvent

dans le chemin d'accès aux classes. Créez vos propres répertoires et définissez le chemin d'accès complet aux classes pour le service de télémétrie (MQXR).

Vous pouvez accroître le chemin d'accès aux classes utilisé par le service de télémétrie (MQXR) en définissant le chemin d'accès aux classes dans le fichier `service.env`. `CLASSPATH` doit être indiqué en majuscules et peut uniquement contenir des littéraux. Vous ne pouvez pas utiliser de variables dans `CLASSPATH` ; par exemple, l'instruction `CLASSPATH=%CLASSPATH%` est incorrecte. Le service de télémétrie (MQXR) définit son propre chemin d'accès aux classes? La variable `CLASSPATH` définie dans le fichier `service.env` est ajoutée à ce dernier.

Le service de télémétrie (MQXR) fournit deux rappels qui renvoient le nom d'utilisateur et le mot de passe pour un client connecté au canal MQTT. Le nom d'utilisateur et le mot de passe sont définis dans l'objet `MqttConnectOptions`. Voir la «Modèle de méthode `JAASLoginModule.Login()`», à la page 374 pour un exemple d'accès au Nom d'utilisateur et au Mot de passe.

### Exemple de fichier `jaas.config`

Exemple de fichier de configuration JAAS avec une configuration nommée `MQXRConfig`

```
MQXRConfig {
samples.JAASLoginModule required debug=true;
//com.ibm.security.auth.module.NTLoginModule required;
//com.ibm.security.auth.module.Krb5LoginModule required
//    principal=principal@your_realm
//    useDefaultCcache=TRUE
//    renewTGT=true;
//com.sun.security.auth.module.NTLoginModule required;
//com.sun.security.auth.module.UnixLoginModule required;
//com.sun.security.auth.module.Krb5LoginModule required
//    useTicketCache="true"
//    ticketCache="${user.home}/${}tickets";
};
```

### Modèle de méthode `JAASLoginModule.Login()`

Exemple de module de connexion JAAS codé pour recevoir le nom d'utilisateur et le mot de passe fournis par un client MQTT.

```
public boolean login()
throws javax.security.auth.login.LoginException {
    javax.security.auth.callback.Callback[] callbacks =
    new javax.security.auth.callback.Callback[2];
    callbacks[0] = new javax.security.auth.callback.NameCallback("NameCallback");
    callbacks[1] = new javax.security.auth.callback.PasswordCallback(
    "PasswordCallback", false);
    try {
        callbackHandler.handle(callbacks);
        String username = ((javax.security.auth.callback.NameCallback) callbacks[0])
        .getName();
        char[] password = ((javax.security.auth.callback.PasswordCallback) callbacks[1])
        .getPassword();
        // Accept everything.
        if (true) {
            loggedIn = true;
        } else
        {
            throw new javax.security.auth.login.FailedLoginException("Login failed");
        }
        principal= new JAASPrincipal(username);
    } catch (java.io.IOException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    } catch (javax.security.auth.callback.UnsupportedCallbackException exception) {
        throw new javax.security.auth.login.LoginException(exception.toString());
    }
}

return loggedIn;
}
```

## Tâches associées

Résolution des problèmes : Le module de connexion JAAS n'est pas appelé par le service de télémétrie

## Référence associée

[Classe MQXR AuthCallback](#)

# Administration d'un client AMQP

Vous pouvez administrer un client AMQP à l'aide de IBM MQ Explorer ou d'une ligne de commande. Utilisez l'explorateur pour configurer des canaux et surveiller les clients AMQP qui sont connectés à IBM MQ. Configurez la sécurité des clients AMQP à l'aide de TLS et de JAAS.

## Avant de commencer

Pour plus d'informations sur l'installation d'AMQP sur votre plateforme, voir [Choix concernant ce qu'il faut installer](#).

## Administration à l'aide de IBM MQ Explorer

Utilisez l'explorateur pour configurer les canaux AMQP et surveiller les clients AMQP qui sont connectés à IBM MQ. Vous pouvez configurer la sécurité des clients AMQP à l'aide de TLS et de JAAS.

## Administration à l'aide de la ligne de commande

Vous pouvez administrer un client AMQP à partir de la ligne de commande à l'aide des commandes MQSC.

## Le service AMQP ne démarre pas automatiquement au démarrage du gestionnaire de files d'attente

Depuis la IBM MQ 9.4.0, le comportement par défaut du paramètre de l'attribut **CONTROL** pour le démarrage du service AMQP a changé. Lors de la création et du démarrage d'un nouveau gestionnaire de files d'attente, le service AMQP ne démarre pas automatiquement dans le cadre du processus de démarrage du gestionnaire de files d'attente.

Entre IBM MQ 9.0.4 et IBM MQ 9.4.0, le comportement par défaut du paramètre de l'attribut **CONTROL** pour le démarrage du service AMQP est QMGR.

Si le composant AMQP a été installé, le service AMQP démarre automatiquement, même s'il n'est pas utilisé. Pour éviter le démarrage par défaut de la machine virtuelle Java (JVM) AMQP, vous disposez de deux options:

- Pas d'installation du composant AMQP, ou
- Remplacement de l'attribut **CONTROL** du service AMQP par MANUAL après le démarrage du gestionnaire de files d'attente.

Depuis la IBM MQ 9.4.0, les gestionnaires de files d'attente nouvellement créés ont rétabli le paramètre de l'attribut **CONTROL** de SYSTEM.AMQP.SERVICE sur MANUAL, qui était le paramètre par défaut avant IBM MQ 9.0.4.

Les gestionnaires de files d'attente migrés, s'ils utilisent AMQP, continuent à démarrer automatiquement le service lors du démarrage du gestionnaire de files d'attente. Pour déterminer si AMQP a été utilisé, les éléments suivants sont vérifiés:

- Canaux AMQP existants
- Messages démarrés du canal dans le journal des erreurs AMQP.



### Avertissement :

- Cela se produit une seule fois ; la première fois que le gestionnaire de files d'attente est démarré après une mise à niveau.
- Lors de la migration, si l'attribut **CONTROL** passe de QMGR à MANUAL, un message d'information est consigné dans le journal des erreurs IBM MQ pour indiquer la modification. Voir

Emplacement des journaux AMQP, des journaux d'erreurs et des fichiers de configuration pour plus d'informations.

Si vous souhaitez que le service AMQP démarre automatiquement, remplacez l'attribut **CONTROL** du service par **QMGR** et redémarrez le gestionnaire de files d'attente. Les redémarrages suivants du gestionnaire de files d'attente démarrent le service AMQP.

## Affichage des objets IBM MQ utilisés par les clients AMQP

Vous pouvez afficher les différentes ressources IBM MQ utilisées par les clients AMQP, par exemple les connexions et les abonnements.

### Connexions

Lorsque le service AMQP est démarré, de nouvelles connexions sont créées et connectées au gestionnaire de files d'attente. Ce pool de connexions Hconns est utilisé lorsque les clients AMQP publient des messages. Vous pouvez afficher les connexions Hconns à l'aide de la commande **DISPLAY CONN**. Exemple :

```
DISPLAY CONN(*) TYPE(CONN) WHERE (APPLDESC LK 'IBM MQ Advanced Message Queuing Protocol')
```

Cette commande affiche également les connexions Hconns spécifiques au client. Les connexions Hconns dont l'attribut d'ID client est vide sont les connexions Hconns utilisées dans le pool

Lorsqu'un client AMQP se connecte à un canal AMQP, un nouveau Hconn est connecté au gestionnaire de files d'attente. Cette connexion Hconn est utilisée pour consommer les messages de manière asynchrone pour les abonnements créés par le client AMQP. Vous pouvez afficher le nom de connexion Hconn utilisé par un client AMQP particulier à l'aide de la commande **DISPLAY CONN**. Exemple :

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_abcd1234')
```

### Abonnements créés par les clients

Lorsqu'un client AMQP s'abonne à une rubrique, un nouvel abonnement IBM MQ est créé. Le nom de l'abonnement inclut les informations suivantes:

- Nom du client. Si le client a rejoint un abonnement partagé, le nom du partage est utilisé
- Modèle de rubrique auquel le client s'est abonné
- Préfixe. Le préfixe est *private* si le client a créé un abonnement non partagé ou *share* si le client a rejoint un abonnement partagé

Pour afficher les abonnements utilisés par un client AMQP particulier, exécutez la commande **DISPLAY SUB** et filtrez sur le préfixe *private* :

```
DISPLAY SUB(':private:*')
```

Pour afficher les abonnements partagés qui sont utilisés par plusieurs clients, exécutez la commande **DISPLAY SUB** et filtrez sur le préfixe *share* :

```
DISPLAY SUB(':share:*')
```

Etant donné que les abonnements partagés peuvent être utilisés par plusieurs clients AMQP, vous pouvez afficher les clients qui consomment actuellement des messages de l'abonnement partagé. Pour ce faire, vous pouvez répertorier les connexions Hconns dont un descripteur est actuellement ouvert dans la file d'attente d'abonnement. Pour afficher les clients qui utilisent actuellement un partage, procédez comme suit:

1. Recherchez le nom de la file d'attente que l'abonnement partagé utilise comme destination. Exemple :



```

DISPLAY SUB(':private:recv_e298452:public') DEST
5 : DISPLAY SUB(':private:recv_e298452:public') DEST
AMQ8096: WebSphere MQ subscription inquired.
SUBID(414D5120514D312020202020202020707E0A565C2D0020)
SUB(:private:recv_e298452:public)
DEST(SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)

```

2. Exécutez la commande **DISPLAY CONN** pour rechercher les descripteurs ouverts sur cette file d'attente:

```

DISPLAY CONN(*) TYPE(HANDLE) WHERE (OBJNAME
EQ SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)
21 : DISPLAY CONN(*) TYPE(HANDLE) WHERE(OBJNAME EQ
SYSTEM.MANAGED.DURABLE.560A7E7020002D5B)

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(HANDLE)

OBJNAME(SYSTEM.BASE.TOPIC) OBJTYPE(TOPIC)

OBJNAME(SYSTEM.MANAGED.DURABLE.560A7E7020002961)
OBJTYPE(QUEUE)

```

3. Pour chacun des descripteurs, affichez l'ID client AMQP dont le descripteur est ouvert:

```

DISPLAY CONN(707E0A56642B0020) CLIENTID
23 : DISPLAY CONN(707E0A56642B0020) CLIENTID

AMQ8276: Display Connection details.
CONN(707E0A56642B0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_8f02c9d)
DISPLAY CONN(707E0A565F290020) CLIENTID
24 : DISPLAY CONN(707E0A565F290020) CLIENTID
AMQ8276: Display Connection details.
CONN(707E0A565F290020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_86d8888)

```

## Identification, autorisation et authentification du client AMQP

Comme les autres applications client IBM MQ, vous pouvez sécuriser les connexions AMQP de différentes manières.

Vous pouvez utiliser les fonctions de sécurité suivantes pour sécuriser les connexions AMQP à IBM MQ:

- [Enregistrements d'authentification de canal](#)
- [Authentification de connexion](#)
- Configuration utilisateur MCA de canal
- Définitions des droits d'accès IBM MQ
- [Connectivité TLS](#)

Du point de vue de la sécurité, l'établissement d'une connexion comprend les deux étapes suivantes:

- Décider si la connexion doit se poursuivre
- Choix de l'identité IBM MQ que l'application suppose pour les vérifications ultérieures des droits d'accès

Les informations suivantes décrivent les différentes configurations IBM MQ et les étapes à suivre lorsqu'un client AMQP tente d'établir une connexion. Toutes les configurations IBM MQ n'utilisent pas toutes les étapes décrites. Par exemple, certaines configurations n'utilisent pas TLS pour les connexions à l'intérieur du pare-feu de la société et d'autres utilisent TLS mais n'utilisent pas de certificats client

pour l'authentification. De nombreux environnements n'utilisent pas de modules JAAS personnalisés ou personnalisés.

## **Etablissement d'une connexion**

Les étapes suivantes décrivent ce qui se passe lorsqu'une connexion est établie par un client AMQP. Les étapes déterminent si la connexion se poursuit et quelle identité IBM MQ l'application utilise pour les vérifications des droits d'accès:

1. Si le client ouvre une connexion TLS à IBM MQ et fournit un certificat, le gestionnaire de files d'attente tente de valider le certificat client.
2. Si le client fournit des données d'identification par nom d'utilisateur et mot de passe, une trame AMQP SASL est reçue par le gestionnaire de files d'attente et la configuration MQ CONNAUTH est vérifiée.
3. Les règles d'authentification de canal MQ sont vérifiées (par exemple, si l'adresse IP et le nom distinctif du certificat TLS sont valides)
4. Le canal MCAUSER est activé, sauf si les règles d'authentification de canal en déterminent le contraire.
5. Si un module JAAS a été configuré, il est appelé
6. MQ Vérification des droits d'accès CONNECT appliquée à l'ID utilisateur MQ résultant.
7. Connexion établie avec une identité IBM MQ supposée.

## **Publication d'un message**

Les étapes suivantes décrivent ce qui se passe lorsqu'un message est publié par un client AMQP. Les étapes déterminent si la connexion se poursuit et quelle identité IBM MQ l'application utilise pour les vérifications des droits d'accès:

1. Le cadre de connexion de lien AMQP arrive dans le gestionnaire de files d'attente. Le droit de publication IBM MQ pour la chaîne de rubrique spécifiée est vérifié pour l'identité de l'utilisateur MQ établie lors de la connexion.
2. Le message est publié dans la chaîne de rubrique spécifiée.

## **Abonnement à un modèle de rubrique**

Les étapes suivantes décrivent ce qui se passe lorsqu'un client AMQP s'abonne à un modèle de rubrique. Les étapes déterminent si la connexion se poursuit et quelle identité IBM MQ l'application utilise pour les vérifications des droits d'accès:

1. Le cadre de connexion de lien AMQP arrive dans le gestionnaire de files d'attente. Le droit d'abonnement IBM MQ pour le modèle de rubrique spécifié est vérifié pour l'identité de l'utilisateur MQ établie lors de la connexion.
2. L'abonnement est créé.

## **Identité et autorisation du client AMQP**

Utilisez l'ID client AMQP, le nom d'utilisateur AMQP ou une identité client commune définie sur le canal ou dans une règle d'authentification de canal pour l'autorisation d'accès aux objets IBM MQ .

L'administrateur fait le choix lors de la définition ou de la modification du canal AMQP, en configurant le paramètre CONNAUTH du gestionnaire de files d'attente ou en définissant des règles d'authentification de canal. L'identité est utilisée pour autoriser l'accès aux rubriques IBM MQ. Le choix se fait en fonction des éléments suivants:

1. Attribut USECLNTID du canal.
2. Attribut ADOPTCTX de la règle CONNAUTH du gestionnaire de files d'attente.
3. Attribut MCAUSER défini sur le canal.
4. Attribut USERSRC d'une règle d'authentification de canal correspondante.

**Eviter les incidents :** L'identité choisie par ce processus est ensuite appelée, par exemple par la commande DISPLAY CHSTATUS (AMQP), MCAUSER du client. Sachez qu'il ne s'agit pas nécessairement de la même identité que le MCAUSER du canal auquel il est fait référence dans le choix (2).

Utilisez la commande IBM MQ **setmqaut** pour sélectionner les objets et les actions qui sont autorisés à être utilisés par l'identité associée au canal AMQP. Par exemple, les commandes suivantes autorisent une identité de canal AMQPClient, fournie par l'administrateur du gestionnaire de files d'attente QM1:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPClient -all +pub +sub
```

et

```
setmqaut -m QM1 -t qmgr -p AMQPClient -all +connect
```

## Authentification du client AMQP à l'aide d'un mot de passe

Authentifiez le nom d'utilisateur du client AMQP à l'aide du mot de passe du client. Vous pouvez authentifier le client à l'aide d'une identité différente de celle utilisée pour autoriser le client à publier des rubriques et à s'y abonner.

Le service AMQP peut utiliser MQ CONNAUTH ou JAAS pour authentifier le nom d'utilisateur du client. Si l'un d'eux est configuré, le mot de passe fourni par le client est vérifié par la configuration MQ CONNAUTH ou le module JAAS .

La procédure suivante décrit des exemples d'étapes permettant d'authentifier des utilisateurs individuels auprès des utilisateurs et des mots de passe du système d'exploitation local et, en cas de réussite, d'adopter l'identité commune AMQPUser:

1. L'administrateur IBM MQ définit l'identité MCAUSER du canal AMQP sur n'importe quel nom, tel que AMQPUser, à l'aide de IBM MQ Explorer.
2. L'administrateur IBM MQ autorise AMQPUser à publier et à s'abonner à n'importe quelle rubrique:

```
setmqaut -m QM1 -t topic -n SYSTEM.BASE.TOPIC -p AMQPUser -all +pub +sub +connect
```

3. L'administrateur IBM MQ configure une règle IDPWOS CONNAUTH pour vérifier le nom d'utilisateur et le mot de passe présentés par le client. La règle CONNAUTH doit définir CHCKCLNT (REQUIRED) et ADOPTCTX (NO).

**Remarque :** Il est recommandé d'utiliser des règles d'authentification de canal et de définir l'attribut de canal MCAUSER sur un utilisateur qui ne dispose pas de privilèges, afin d'autoriser un contrôle plus important sur les connexions au gestionnaire de files d'attente.

## Confidentialité des publications sur les canaux

La confidentialité des publications AMQP envoyées dans les deux sens sur les canaux AMQP est sécurisée à l'aide de TLS pour chiffrer les transmissions sur la connexion.

Les clients AMQP qui se connectent aux canaux AMQP utilisent TLS pour sécuriser la confidentialité des publications transmises sur le canal à l'aide de la cryptographie à clé symétrique. Etant donné que les noeuds finaux ne sont pas authentifiés, vous ne pouvez pas accréditer le chiffrement de canal seul. Combinez la sécurisation de la confidentialité avec l'authentification du serveur ou l'authentification mutuelle.

Au lieu d'utiliser TLS, certains types de réseau privé virtuel (VPN), tels que IPsec, authentifient les noeuds finaux d'une connexion TCP/IP. Le réseau privé virtuel VPN chiffre chaque paquet transitant sur le réseau. Une fois qu'une telle connexion VPN est établie, vous avez établi un réseau sécurisé. Vous pouvez connecter des clients AMQP à des canaux AMQP à l'aide de TCP/IP sur le réseau VPN.

Le chiffrement des connexions TLS sans authentification du serveur expose la connexion aux attaques man-in-the-middle. Bien que les informations que vous échangez soient protégées contre les écoutes

clandestines, vous ne savez pas avec qui vous réalisez les échanges. A moins que vous ne contrôliez le réseau, vous êtes exposé à ce que quelqu'un intercepte vos transmissions IP, en se faisant passer pour un noeud final.

Vous pouvez créer une connexion TLS chiffrée, sans authentifier le serveur, à l'aide d'un CipherSpec d'échange de clés Diffie-Hellman qui prend en charge le protocole TLS anonyme. Le secret maître, partagé entre le client et le serveur et utilisé pour chiffrer les transmissions TLS, est établi sans échanger de certificat de serveur signé de manière privée.

Etant donné que les connexions anonymes ne sont pas sécurisées, la plupart des implémentations TLS n'utilisent pas par défaut les CipherSpecs anonymes. Si une demande client de connexion TLS est acceptée par un canal AMQP, le canal doit disposer d'un magasin de clés protégé par une phrase passe. Par défaut, les implémentations TLS n'utilisant pas de CipherSpecs anonyme, le magasin de clés doit contenir un certificat signé de manière privée que le client peut authentifier.

Si vous utilisez des spécifications CipherSpecs anonymes, le magasin de clés du serveur doit exister, mais il ne doit pas forcément contenir des certificats signés de manière privée.

Une autre façon d'établir une connexion cryptée consiste à remplacer le fournisseur d'accréditation côté client par votre propre implémentation. Votre fournisseur d'accréditation n'effectuerait pas l'authentification du certificat serveur, mais la connexion serait cryptée.

## Configuration des clients AMQP avec TLS

Vous pouvez configurer les clients AMQP pour qu'ils utilisent TLS afin de protéger les données circulant sur le réseau et d'authentifier l'identité du gestionnaire de files d'attente auquel le client se connecte.

Pour utiliser TLS pour la connexion entre un client AMQP et un canal AMQP, vous devez vous assurer que le gestionnaire de files d'attente a été configuré pour TLS. La rubrique [Configuration de TLS sur les gestionnaires de files d'attente](#) explique comment configurer le magasin de clés à partir duquel un gestionnaire de files d'attente lit les certificats TLS.


Lorsque le gestionnaire de files d'attente a été configuré avec un magasin de clés, vous devez configurer les attributs TLS sur le canal AMQP auquel les clients se connecteront. Les canaux AMQP ont quatre attributs liés à la configuration TLS comme suit:

### SSLCAUTH

L'attribut SSLCAUTH permet d'indiquer si le gestionnaire de files d'attente doit exiger qu'un client AMQP présente un certificat client pour vérifier son identité.

### SSLCIPH

L'attribut SSLCIPH spécifie le chiffrement que le canal doit utiliser pour coder les données dans le flux TLS.

 Depuis IBM MQ 9.4.0, les canaux AMQP prennent en charge les CipherSpecs génériques ANY\*. Pour plus d'informations sur les CipherSpecs, voir [Activation des CipherSpecs](#).

### SSLPEER

L'attribut SSLPEER est utilisé pour spécifier le nom distinctif (DN) auquel un certificat client doit correspondre si une connexion doit être autorisée.

### CERTLABL

Le certificat CERTLABL indique le certificat que le gestionnaire de files d'attente doit présenter au client. Le magasin de clés du gestionnaire de files d'attente peut contenir plusieurs certificats. Cet attribut vous permet de spécifier le certificat à utiliser pour les connexions à ce canal. Si aucun CERTLABL n'est spécifié, le certificat du référentiel de clés du gestionnaire de files d'attente dont le libellé correspond à l'attribut CERTLABL du gestionnaire de files d'attente est utilisé.

Une fois que vous avez configuré votre canal AMQP avec les attributs TLS, vous devez redémarrer le service AMQP à l'aide de la commande suivante:

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

Lorsqu'un client AMQP se connecte à un canal AMQP protégé par TLS, le client vérifie l'identité du certificat présenté par le gestionnaire de files d'attente. Pour ce faire, vous devez configurer votre client AMQP avec un magasin de clés de confiance contenant le certificat du gestionnaire de files d'attente. Les étapes à suivre varient en fonction du client AMQP que vous utilisez. Pour plus d'informations sur les différents clients et API AMQP, voir la documentation correspondante du client AMQP.

#### Référence associée

[DEFINE CHANNEL](#) (définition d'un nouveau canal)

[STOP SERVICE](#) (arrêt d'un service) sur Multiplatforms

[START SERVICE](#) (démarrage d'un service) sur Multiplatforms

## Déconnexion des clients AMQP du gestionnaire de files d'attente

Si vous souhaitez déconnecter des clients AMQP du gestionnaire de files d'attente, exécutez la commande `PURGE CHANNEL` ou arrêtez la connexion au client AMQP.

- Exécutez la commande **`PURGE CHANNEL`**. Exemple :

```
PURGE CHANNEL (MYAMQP) CLIENTID('recv_28dbb7e')
```

- Vous pouvez également arrêter la connexion que le client AMQP utilise pour déconnecter le client en procédant comme suit:

- Recherchez la connexion utilisée par le client en exécutant la commande **`DISPLAY CONN`**. Exemple :

```
DISPLAY CONN(*) TYPE(CONN) WHERE (CLIENTID EQ 'recv_28dbb7e')
```

La sortie de la commande est la suivante:

```
DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
40 : DISPLAY CONN(*) TYPE(CONN) WHERE(CLIENTID EQ 'recv_28dbb7e')
AMQ8276: Display Connection details.
CONN(707E0A565F2D0020)
EXTCONN(414D5143514D31202020202020202020)
TYPE(CONN)
CLIENTID(recv_28dbb7e)
```

- Arrêtez la connexion. Exemple :

```
STOP CONN(707E0A565F2D0020)
```

## Administration de la multidiffusion

Utilisez ces informations pour en savoir plus sur les tâches d'administration de multidiffusion IBM MQ telles que la réduction de la taille des messages de multidiffusion et l'activation de la conversion de données.

### Initiation à la multidiffusion

Utilisez ces informations pour vous initier aux rubriques de multidiffusion IBM MQ et aux objets d'informations de communication.

#### Pourquoi et quand exécuter cette tâche

La messagerie IBM MQ Multicast utilise le réseau pour distribuer des messages en mappant des rubriques à des adresses de groupe. Les tâches suivantes permettent de tester rapidement si l'adresse IP et le port requis sont correctement configurés pour la messagerie multidiffusion.

### Création d'un objet COMMINFO pour la multidiffusion

L'objet informations de communication (COMMINFO) contient les attributs associés à la transmission multidiffusion. Pour plus d'informations sur les paramètres de l'objet COMMINFO, voir [DEFINE COMMINFO](#).

Utilisez l'exemple de ligne de commande suivant pour définir un objet COMMINFO pour la multidiffusion:

```
DEFINE COMMINFO(MC1) GRPADDR(group address) PORT(port number)
```

où *MC1* est le nom de votre objet COMMINFO, *adresse de groupe* est l'adresse IP de multidiffusion de votre groupe ou le nom DNS, et le *numéro de port* est le port sur lequel effectuer la transmission (la valeur par défaut est 1414).

Un nouvel objet COMMINFO appelé *MC1* est créé. Il s'agit du nom que vous devez spécifier lors de la définition d'un objet TOPIC dans l'exemple suivant.

### Création d'un objet TOPIC pour la multidiffusion

Une rubrique est l'objet des informations publiées dans un message de publication/abonnement, et une rubrique se définit en créant un objet TOPIC. Les objets TOPIC ont deux paramètres qui définissent s'ils peuvent être utilisés avec la multidiffusion ou non. Ces paramètres sont: **COMMINFO** et **MCAST**.

- **COMMINFO** Ce paramètre indique le nom de l'objet d'information de communication multidiffusion. Pour plus d'informations sur les paramètres de l'objet COMMINFO, voir [DEFINE COMMINFO](#).
- **MCAST** Ce paramètre indique si la multidiffusion est autorisée à ce niveau dans l'arborescence des rubriques.

Utilisez l'exemple de ligne de commande suivant pour définir un objet TOPIC pour la multidiffusion:

```
DEFINE TOPIC(ALLSPORTS) TOPICSTR('Sports') COMMINFO(MC1) MCAST(ENABLED)
```

Un nouvel objet TOPIC appelé *ALLSPORTS* est créé. Il comporte une chaîne de rubrique *Sports*, son objet d'informations de communication associé est appelé *MC1* (nom que vous avez spécifié lors de la définition d'un objet COMMINFO dans l'exemple précédent) et la multidiffusion est activée.

### Test de la publication / abonnement multidiffusion

Une fois les objets TOPIC et COMMINFO créés, ils peuvent être testés à l'aide de l'exemple `amqspubc` et de l'exemple `amqssubc`. Pour plus d'informations sur ces exemples, voir [Exemples de programmes de publication / abonnement](#).

1. Ouvrez deux fenêtres de ligne de commande. La première ligne de commande concerne l'exemple de publication `amqspubc` et la deuxième ligne de commande concerne l'exemple d'abonnement `amqssubc`.
2. Entrez la commande suivante sur la ligne de commande 1:

```
amqspubc Sports QM1
```

où *Sports* est la chaîne de rubrique de l'objet TOPIC défini dans un exemple précédent, et *QM1* est le nom du gestionnaire de files d'attente.

3. Entrez la commande suivante sur la ligne de commande 2:

```
amqssubc Sports QM1
```

où *Sports* et *QM1* sont identiques à ceux utilisés à l'étape «2», à la page 382.

4. Entrez `Hello world` sur la ligne de commande 1. Si le port et l'adresse IP spécifiés dans l'objet COMMINFO sont configurés correctement, l'exemple `amqssubc`, qui écoute sur le port les publications à partir de l'adresse spécifiée, génère `Hello world` sur la ligne de commande 2.

## Topologie de rubrique de multidiffusion IBM MQ

Utilisez cet exemple pour comprendre la topologie de rubrique IBM MQ Multicast.

La prise en charge de la multidiffusion IBM MQ requiert que chaque sous-arborescence ait son propre groupe de multidiffusion et son propre flux de données au sein de la hiérarchie totale.

Le schéma d'adressage IP *réseau avec classe* désigne un espace adresse pour l'adresse de multidiffusion. La plage de multidiffusion complète de l'adresse IP est 224 . 0 . 0 . 0 à 239 . 255 . 255 . 255, mais certaines de ces adresses sont réservées. Pour la liste des adresses réservées, prenez contact avec votre administrateur système ou consultez la page <https://www.iana.org/assignments/multicast-addresses> pour plus d'informations. Il est recommandé d'utiliser l'adresse de multidiffusion locale dans la plage de 239 . 0 . 0 . 0 à 239 . 255 . 255 . 255.

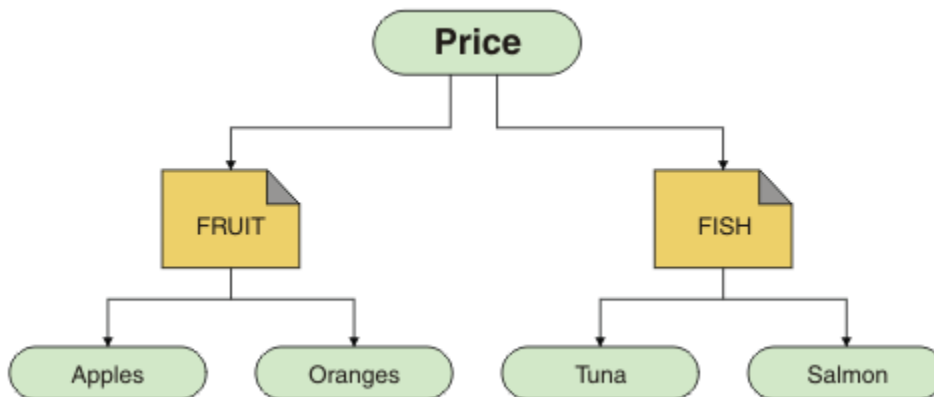
Dans le diagramme suivant, il existe deux flux de données de multidiffusion possibles:

```
DEF COMMINFO(MC1) GRPADDR(239.XXX.XXX.XXX
)
DEF COMMINFO(MC2) GRPADDR(239.YYY.YYY.YYY)
```

où 239 . XXX . XXX . XXX et 239 . YYY . YYY . YYY sont des adresses de multidiffusion valides.

Ces définitions de rubrique sont utilisées pour créer une arborescence de rubriques, comme illustré dans le diagramme suivant:

```
DEFINE TOPIC(FRUIT) TOPICSTRING('Price/FRUIT') MCAST(ENABLED) COMMINFO(MC1)
DEFINE TOPIC(FISH) TOPICSTRING('Price/FISH') MCAST(ENABLED) COMMINFO(MC2)
```



Chaque objet d'informations de communication multidiffusion (COMMINFO) représente un flux de données différent car leurs adresses de groupe sont différentes. Dans cet exemple, la rubrique FRUIT est définie pour utiliser l'objet COMMINFO MC1, la rubrique FISH est définie pour utiliser l'objet COMMINFO MC2 et le nœud Price n'a pas de définitions de multidiffusion.

IBM MQ La multidiffusion est limitée à 255 caractères pour les chaînes de rubrique. Cette limitation signifie que les noms des nœuds et des nœuds feuille doivent être pris en compte dans l'arborescence. Si les noms des nœuds et des nœuds feuille sont trop longs, la chaîne de rubrique peut dépasser 255 caractères et renvoyer le code anomalie 2425 (0979) (RC2425): MQRC\_TOPIC\_STRING\_ERROR. Il est recommandé de rendre les chaînes de rubrique aussi courtes que possible car des chaînes de rubrique plus longues peuvent avoir un effet négatif sur les performances.

## Contrôle de la taille des messages de multidiffusion

Utilisez ces informations pour en savoir plus sur le format de message IBM MQ et réduire la taille des messages IBM MQ.

Les messages IBM MQ sont associés à un certain nombre d'attributs qui sont contenus dans le descripteur de message. Pour les messages de petite taille, ces attributs peuvent représenter la majeure partie du trafic de données et peuvent avoir un effet négatif significatif sur le débit de transmission. IBM MQ Multicast permet à l'utilisateur de configurer les attributs qui, le cas échéant, sont transmis avec le message.

La présence d'attributs de message, autres que la chaîne de rubrique, varie selon que l'objet COMMINFO indique qu'ils doivent être envoyés ou non. Si un attribut n'est pas transmis, l'application de réception applique une valeur par défaut. Les valeurs MQMD par défaut ne sont pas nécessairement identiques à la valeur MQMD\_DEFAULT et sont décrites dans [Tableau 19](#), à la page 384.

L'objet COMMINFO contient l'attribut MCPROP qui contrôle le nombre de zones MQMD et de propriétés utilisateur qui circulent avec le message. En définissant la valeur de cet attribut sur un niveau approprié, vous pouvez contrôler la taille des messages de multidiffusion IBM MQ :

#### **MCPROP**

Les propriétés de multidiffusion contrôlent la quantité de propriétés MQMD et propriétés utilisateur émises avec le message.

#### **TOUT**

Toutes les propriétés utilisateur et toutes les zones du MQMD sont transmises.

#### **REPONSE**

Seules les propriétés utilisateur et les zones MQMD liées à la réponse des messages sont transmises. Ces propriétés sont :

- MsgType
- MessageId
- CorrelId
- ReplyToQ
- ReplyToQmgr

#### **UTILISATEUR**

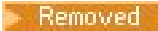
Seules les propriétés utilisateur sont transmises.

#### **Aucun**

Aucune propriété utilisateur ni zone MQMD n'est transmise.

#### **COMPAT**

Cette valeur entraîne la transmission du message dans un mode compatible à RMM, ce qui permet une certaine interopérabilité avec les applications XMS et IBM Integration Bus RMM en cours.

 XMS .NET La messagerie multidiffusion (à l'aide de RMM) a été dépréciée dans IBM MQ 9.2 et a été retirée dans IBM MQ 9.3.

### **Attributs de message de multidiffusion**

Les attributs de message peuvent provenir de différents emplacements, tels que MQMD, les zones de MQRFH2 et les propriétés de message.

Le tableau suivant montre ce qui se passe lorsque des messages sont envoyés en fonction de la valeur de MCPROP (décrite précédemment dans cette section) et de la valeur par défaut utilisée lorsqu'un attribut n'est pas envoyé.

<i>Tableau 19. Attributs de messagerie et comment ils sont liés à la multidiffusion</i>		
<b>Attribut</b>	<b>Action lors de l'utilisation de la multidiffusion</b>	<b>Valeur par défaut si non transmise</b>
TopicString	Toujours inclus	Non applicable
MQMQ StrucId	Non transmis	Non applicable
Version MQMD	Non transmis	Non applicable



Tableau 19. Attributs de messagerie et comment ils sont liés à la multidiffusion (suite)

Attribut	Action lors de l'utilisation de la multidiffusion	Valeur par défaut si non transmise
Rapport	Inclus si non par défaut	0
MsgType	Inclus si non par défaut	MQMT_DATAGRAM
Expiration	Inclus si non par défaut	0
Commentaires	Inclus si non par défaut	0
Codage	Inclus si non par défaut	MQENC_NORMAL (équivalent)
CodedCharSetId	Inclus si non par défaut	1208
Format	Inclus si non par défaut	MQRFH2
Priorité	Inclus si non par défaut	4
Persistance	Inclus si non par défaut	MQPER_NON_PERSISTENT
MsgId	Inclus si non par défaut	NULL
CorrelId	Inclus si non par défaut	NULL
BackoutCount	Inclus si non par défaut	0
ReplyToQ	Inclus si non par défaut	Blanc
ReplyToQMgr	Inclus si non par défaut	Blanc
UserIdentifier	Inclus si non par défaut	Blanc
AccountingToken	Inclus si non par défaut	NULL
PutAppIType	Inclus si non par défaut	MQAT_JAVA
PutAppIName	Inclus si non par défaut	Blanc
PutDate	Inclus si non par défaut	Blanc
PutTime	Inclus si non par défaut	Blanc
ApplOriginData	Inclus si non par défaut	Blanc
GroupID	Exclu	Non applicable
MsgSeqNumber	Exclu	Non applicable
Décalage	Exclu	Non applicable
MsgFlags	Exclu	Non applicable
OriginalLength	Exclu	Non applicable
UserProperties	Inclus	Non applicable

#### Référence associée

 ALTER COMMINFO  
DEFINI COMMINFO

## Activation de la conversion de données pour la messagerie multidiffusion

Utilisez ces informations pour comprendre le fonctionnement de la conversion de données pour la messagerie IBM MQ Multicast.

IBM MQ Multicast est un protocole partagé et sans connexion. Par conséquent, il n'est pas possible pour chaque client d'effectuer des demandes spécifiques de conversion de données. Chaque client abonné au même flux de multidiffusion reçoit les mêmes données binaires ; par conséquent, si une conversion de données IBM MQ est requise, la conversion est effectuée localement sur chaque client.

Dans une installation à plateforme mixte, il se peut que la plupart des clients requièrent les données dans un format qui n'est pas le format natif de l'application de transmission. Dans ce cas, les valeurs **CCSID** et **ENCODING** de l'objet COMMINFO de multidiffusion peuvent être utilisées pour définir le codage de la transmission de message à des fins d'efficacité.

IBM MQ Multicast prend en charge la conversion de données de la charge de message pour les formats intégrés suivants:

- MQADMIN
- MQEVENT
- MQPCF
- MQRFH
- MQRFH2
- MQSTR

En plus de ces formats, vous pouvez également définir vos propres formats et utiliser un exit de conversion de données [MQDXP-Paramètre d'exit de conversion de données](#) .

Pour plus d'informations sur la programmation des conversions de données, voir [Conversion de données dans MQI pour la messagerie multidiffusion](#).

Pour plus d'informations sur la conversion de données, voir [Conversion de données](#).

Pour plus d'informations sur les exits de conversion de données et ClientExitPath, voir la section [ClientExitPath](#) du fichier de configuration client.

## Surveillance des applications de multidiffusion

Utilisez ces informations pour en savoir plus sur l'administration et la surveillance de la multidiffusion IBM MQ .

L'état des diffuseurs actuels et des abonnés pour le trafic multidiffusion (par exemple, le nombre de messages envoyés et reçus, ou le nombre de messages perdus) est périodiquement transmis au serveur depuis le client. Lorsque le statut est reçu, l'attribut **COMMEV** de l'objet COMMINFO indique si le gestionnaire de files d'attente insère ou non un message d'événement dans SYSTEM.ADMIN.PUBSUB.EVENT. Le message d'événement contient les informations de statut reçues. Ces informations sont un outil de diagnostic précieux pour trouver la source d'un problème.

Utilisez la commande MQSC **DISPLAY CONN** pour afficher les informations de connexion relatives aux applications connectées au gestionnaire de files d'attente. Pour plus d'informations sur la commande **DISPLAY CONN** , voir [DISPLAY CONN](#).

Utilisez la commande MQSC **DISPLAY TPSTATUS** pour afficher le statut de vos diffuseurs de publications et de vos abonnés. Pour plus d'informations sur la commande **DISPLAY TPSTATUS** , voir [DISPLAY TPSTATUS](#).

### COMMEV et l'indicateur de fiabilité des messages multidiffusion

L' *indicateur de fiabilité*, utilisé conjointement avec l'attribut **COMMEV** de l'objet COMMINFO, est un élément clé de la surveillance des diffuseurs de publications et des abonnés IBM MQ Multicast. L'indicateur de fiabilité (zone **MSGREL** renvoyée dans les commandes de statut de publication ou d'abonnement) est un indicateur IBM MQ qui illustre le pourcentage de transmissions qui ne comportent pas d'erreurs. Parfois, les messages doivent être retransmis en raison d'une erreur de transmission, ce qui se reflète dans la valeur de **MSGREL**. Les causes potentielles des erreurs de transmission incluent les abonnés lents, les réseaux occupés et les indisponibilités du réseau. **COMMEV** contrôle si les messages d'événement sont générés pour les descripteurs de multidiffusion créés à l'aide de l'objet COMMINFO et sont définis sur l'une des trois valeurs possibles:

## DESACTIVE

Les messages d'événement ne sont pas écrits.

## Activée

Les messages d'événement sont toujours écrits, avec une fréquence définie dans le paramètre **COMMINFO MONINT**.

## EXCEPTION

Les messages d'événement sont écrits si la fiabilité des messages est inférieure au seuil de fiabilité. Un niveau de fiabilité de message inférieur ou égal à 90% indique qu'il peut y avoir un problème avec la configuration réseau ou qu'une ou plusieurs applications de publication / abonnement s'exécutent trop lentement:

- La valeur **MSGREL (100, 100)** indique qu'il n'y a eu aucun problème à court terme ou à long terme.
- La valeur **MSGREL (80, 60)** indique que 20% des messages présentent actuellement des problèmes, mais qu'il s'agit également d'une amélioration par rapport à la valeur à long terme de 60.

Les clients peuvent continuer à transmettre et à recevoir du trafic multidiffusion même lorsque la connexion monodiffusion au gestionnaire de files d'attente est interrompue. Par conséquent, les données peuvent être obsolètes.

## Fiabilité des messages de multidiffusion

Utilisez ces informations pour apprendre à définir l'abonnement à la multidiffusion IBM MQ et l'historique des messages.

Un élément clé pour surmonter l'échec de transmission avec multidiffusion est la mise en mémoire tampon des données transmises (historique des messages à conserver à l'extrémité émettrice de la liaison) par IBM MQ. Ce processus signifie qu'aucune mise en mémoire tampon des messages n'est requise dans le processus d'application d'insertion car IBM MQ fournit la fiabilité. La taille de cet historique est configurée via l'objet d'informations de communication (COMMINFO), comme décrit dans les informations suivantes. Un tampon de transmission plus grand signifie qu'il y a plus d'historique de transmission à retransmettre si nécessaire, mais en raison de la nature de la multidiffusion, une distribution assurée à 100% ne peut pas être prise en charge.

L'historique des messages de multidiffusion IBM MQ est contrôlé dans l'objet d'informations de communication (COMMINFO) par l'attribut **MSGHIST** :

### MSGHIST

Cette valeur correspond à la quantité d'historique des messages, en kilooctets, qui est conservée par le système pour gérer les retransmissions dans le cas des accusés de réception négatifs.

La valeur 0 indique le niveau de fiabilité le moins élevé. La valeur par défaut est 100 Ko.

L'historique des nouveaux abonnements IBM MQ Multicast est contrôlé dans l'objet d'informations de communication (COMMINFO) par l'attribut **NSUBHIST** :

### NSUBHIST

L'historique du nouvel abonné indique si un abonné rejoignant un flot de publication reçoit autant de données que disponible, ou reçoit uniquement les publications effectuées à partir du moment de l'abonnement.

#### Aucun

La valeur NONE permet à l'émetteur de transmettre uniquement la publication effectuée à partir du moment de l'abonnement. NONE est la valeur utilisée par défaut.

#### TOUT

La valeur ALL permet à l'émetteur de retransmettre autant d'historique de la rubrique que cela est connu. Dans certaines circonstances, cette situation peut donner un comportement similaire à celui des publications conservées.

**Remarque :** L'utilisation de la valeur de ALL peut avoir un impact négatif sur les performances s'il existe un historique de sujet volumineux car tout l'historique de sujet est retransmis.

## Référence associée

DEFINI COMMINFO

Multi

ALTER COMMINFO

## Tâches de multidiffusion avancées

Utilisez ces informations pour en savoir plus sur les tâches d'administration IBM MQ Multicast avancées telles que la configuration des fichiers `.ini` et l'interopérabilité avec IBM MQ LLM.

Pour plus d'informations sur la sécurité dans une installation de multidiffusion, voir [Sécurité de multidiffusion](#).

## Pontage entre les domaines de publication / abonnement multidiffusion et non multidiffusion

Utilisez ces informations pour comprendre ce qui se passe lorsqu'un diffuseur de publications non multidiffusion est publié dans une rubrique activée pour la multidiffusion IBM MQ.

Si un diffuseur de publications non multidiffusion est publié dans une rubrique définie comme **MCAST** activée et **BRIDGE** activée, le gestionnaire de files d'attente transmet le message en mode multidiffusion directement aux abonnés qui peuvent être en mode écoute. Un diffuseur de publications multidiffusion ne peut pas publier dans des rubriques qui ne sont pas activées pour la multidiffusion.

Les rubriques existantes peuvent être activées en mode multidiffusion en définissant les paramètres **MCAST** et **COMMINFO** d'un objet de rubrique. Pour plus d'informations sur ces paramètres, voir [Concepts de multidiffusion initiaux](#).

L'attribut **BRIDGE** de l'objet COMMINFO contrôle les publications des applications qui n'utilisent pas la multidiffusion. Si **BRIDGE** est défini sur **ENABLED** et que le paramètre **MCAST** de la rubrique est également défini sur **ENABLED**, les publications des applications qui n'utilisent pas la multidiffusion sont routées vers les applications qui le font. Pour plus d'informations sur le paramètre **BRIDGE**, voir [DEFINE COMMINFO](#).

## Configuration des fichiers .ini pour la multidiffusion

Utilisez ces informations pour comprendre les zones de multidiffusion IBM MQ dans les fichiers `.ini`.

Une configuration IBM MQ Multicast supplémentaire peut être effectuée dans un fichier `ini`. Le fichier `ini` spécifique que vous devez utiliser dépend du type d'application:

- Client: configurez le fichier `MQ_DATA_PATH/mqclient.ini`.
- Gestionnaire de files d'attente: configurez le fichier `MQ_DATA_PATH/qmgrs/QMNAME/qm.ini`.

où `MQ_DATA_PATH` est l'emplacement du répertoire de données IBM MQ (`/var/mqm/mqclient.ini`) et `QMNAME` est le nom du gestionnaire de files d'attente auquel s'applique le fichier `.ini`.

Le fichier `.ini` contient des zones utilisées pour affiner le comportement de IBM MQ Multicast:

```
Multicast:
Protocol      = IP | UDP
IPVersion     = IPv4 | IPv6 | ANY | BOTH
LimitTransRate = DISABLED | STATIC | DYNAMIC
TransRateLimit = 100000
SocketTTL    = 1
Batch        = NO
Loop        = 1
Interface    = <IPAddress>
FeedbackMode = ACK | NACK | WAIT1
HeartbeatTimeout = 20000
HeartbeatInterval = 2000
```

## Protocole

### UDP

Dans ce mode, les paquets sont envoyés à l'aide du protocole UDP. Les éléments de réseau ne peuvent pas fournir d'assistance dans la distribution multidiffusion comme ils le font en mode IP. Le format de paquet reste compatible avec PGM. Il s'agit de la valeur par défaut.

### Adresse IP

Dans ce mode, l'émetteur envoie des paquets IP bruts. Les éléments de réseau avec prise en charge de PGM aident à la distribution de paquets multidiffusion fiable. Ce mode est entièrement compatible avec la norme PGM.

## IPVersion

### IPv4

Communiquez uniquement à l'aide du protocole IPv4 . Il s'agit de la valeur par défaut.

### IPv6

Communiquez uniquement à l'aide du protocole IPv6 .

### ANY

Communiquez à l'aide de IPv4, de IPv6 ou des deux, selon le protocole disponible.

### LES DEUX

Prend en charge la communication à l'aide de IPv4 et de IPv6.

## LimitTransDébit

### DEACTIVE

Il n'y a pas de contrôle du débit de transmission. Il s'agit de la valeur par défaut.

### STATIQUE

Implémente le contrôle du débit de transmission statique. L'émetteur ne transmet pas à un débit supérieur au débit spécifié par le paramètre de limite TransRate.

### DYNAMIQUE

L'émetteur adapte son débit de transmission en fonction de la rétroaction qu'il reçoit des récepteurs. Dans ce cas, la limite de débit de transmission ne peut pas être supérieure à la valeur spécifiée par le paramètre TransRateLimit. L'émetteur tente d'atteindre un débit de transmission optimal.

## Limite TransRate

Limite de débit de transmission en kbit/s.

## SocketTTL

La valeur de SocketTTL détermine si le trafic de multidiffusion peut passer par un routeur, ou le nombre de routeurs qu'il peut traverser.

## Lot

Contrôle si les messages sont envoyés par lots ou immédiatement. Il existe 2 valeurs possibles:

- *NON* Les messages ne sont pas envoyés par lots, ils sont envoyés immédiatement.
- *OUI* Les messages sont traités par lots.

## Boucle

Définissez la valeur sur 1 pour activer la boucle de multidiffusion. La boucle de multidiffusion définit si les données envoyées sont ou non rebouclées sur l'hôte.

## Interface

Adresse IP de l'interface sur laquelle transite le trafic multidiffusion. Pour plus d'informations et pour résoudre les problèmes, voir: [Test d'applications multidiffusion sur un réseau non multidiffusion et Définition du réseau approprié pour le trafic multidiffusion](#)

## FeedbackMode

### accusé de réception négatif

Commentaires en retour par des accusés de réception négatifs. Il s'agit de la valeur par défaut.

### accusé de réception

Commentaires en retour par des accusés de réception positifs.

## WAIT1

Rétroaction par des accusés de réception positifs où l'émetteur attend seulement 1 ACK de n'importe lequel des récepteurs.

## HeartbeatTimeout

Délai d'attente du signal de présence en millisecondes. La valeur 0 indique que les événements de délai de signal de présence ne sont pas émis par le ou les récepteurs de la rubrique. La valeur par défaut est 20000.

## HeartbeatInterval

Intervalle des pulsations en millisecondes. La valeur 0 indique qu'aucun signal de présence n'est envoyé. L'intervalle des pulsations doit être considérablement inférieur à la valeur **HeartbeatTimeout** pour éviter les faux événements de délai d'attente des pulsations. La valeur par défaut est 2000.

## Interopérabilité multidiffusion avec la messagerie à faible temps d'attente IBM MQ

Utilisez ces informations pour comprendre l'interopérabilité entre IBM MQ Multicast et IBM MQ Low Latency Messaging (LLM).

Le transfert de contenu de base est possible pour une application utilisant LLM, avec une autre application utilisant la multidiffusion pour échanger des messages dans les deux sens. Bien que la multidiffusion utilise la technologie LLM, le produit LLM lui-même n'est pas intégré. Par conséquent, il est possible d'installer LLM et IBM MQ Multicast, et d'utiliser et de traiter les deux produits séparément.

Les applications LLM qui communiquent avec la multidiffusion peuvent avoir besoin d'envoyer et de recevoir des propriétés de message. Les propriétés de message IBM MQ et les zones MQMD sont transmises en tant que propriétés de message LLM avec des codes de propriété de message LLM spécifiques, comme indiqué dans le tableau suivant:

IBM MQ Propriété	IBM MQ Type de propriété LLM	Type de propriété LLM	Code de propriété LLM
MQMD.Report	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1001
MQMD.MsgType	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1002
MQMD.Expiry	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1003
MQMD.Feedback	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1004
MQMD.Encoding	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1005
MQMD.CodedCharSetId	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1006
MQMD.Format	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_Chaîne	-1007
MQMD.Priority	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1008
MQMD.Persistence	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1009
MQMD.MsgId	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_ByteArray	-1010
MQMD.BackoutCount	RMM_MSG_PROP_INT32	LLM_PROP_KIND_Int32	-1012
MQMD.ReplyToQ	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_Chaîne	-1013
MQMD.ReplyToQMger	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_Chaîne	-1014
MQMD.PutDate	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_Chaîne	-1020
MQMD.PutTime	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_Chaîne	-1021
MQMD.ApplOriginData	RMM_MSG_PROP_BYTES	LLM_PROP_KIND_Chaîne	-1022

Tableau 20. Mappages des propriétés de message IBM MQ vers les propriétés LLM IBM MQ (suite)

IBM MQ Propriété	IBM MQ Type de propriété LLM	Type de propriété LLM	Code de propriété LLM
MQPubOptions	RMM_MSG_PROP_INT32	LLM_PROP_KIND_int32	-1053

Pour plus d'informations sur LLM, voir la documentation du produit LLM: [IBM MQ Low Latency Messaging](#).

## IBM i Administration de IBM MQ for IBM i

Les commandes CL constituent la méthode recommandée pour administrer IBM MQ sur IBM i. Vous pouvez également utiliser les commandes MQSC, les commandes PCF, les commandes de contrôle et l'administration à distance.

### Pourquoi et quand exécuter cette tâche

Les tâches d'administration incluent la création, le démarrage, la modification, l'affichage, l'arrêt et la suppression de clusters, de processus et d'objets IBM MQ (gestionnaires de files d'attente, files d'attente, listes de noms, définitions de processus, canaux, canaux de connexion client, programmes d'écoute, services et objets d'informations d'authentification).

Pour plus de détails sur l'administration de IBM MQ for IBM i, voir les liens suivants.

#### Concepts associés

[Description des noms de bibliothèque de gestionnaire de files d'attente IBM MQ for IBM i](#)

[Services et composants installables sous IBM i](#)

#### Tâches associées

[Modification des informations de configuration IBM MQ sur Multiplatforms](#)

[Configuration de la sécurité sous IBM i](#)

«Appel du gestionnaire de files d'attente de rebut sous IBM i», à la page 174

Sous IBM MQ for IBM i, vous appelez le gestionnaire DLQ en définissant la commande **STRMQMDLQ**.

[Identification des problèmes liés aux applications IBM MQ for IBM i](#)

#### Référence associée

[Objets système et par défaut](#)

## IBM i Gestion d'IBM MQ for IBM i à l'aide de commandes CL

Utilisez ces informations pour comprendre les commandes IBM MQ IBM i.

La plupart des groupes de commandes IBM MQ, y compris ceux associés aux gestionnaires de files d'attente, aux files d'attente, aux rubriques, aux canaux, aux listes de noms, aux définitions de processus et aux objets d'informations d'authentification, sont accessibles à l'aide de la commande **WRK\*** appropriée.

La commande principale de l'ensemble est **WRKMQM**. Cette commande permet, par exemple, d'afficher la liste de tous les gestionnaires de files d'attente du système, ainsi que des informations de statut. Vous pouvez également traiter toutes les commandes spécifiques au gestionnaire de files d'attente à l'aide de différentes options pour chaque entrée.

A partir de la commande **WRKMQM**, vous pouvez sélectionner des zones spécifiques de chaque gestionnaire de files d'attente, par exemple, en utilisant des canaux, des rubriques ou des files d'attente, et à partir de là, sélectionner des objets individuels.

### Enregistrement des définitions d'application IBM MQ

Lorsque vous créez ou personnalisez des applications IBM MQ, il est utile de conserver un enregistrement de toutes les définitions IBM MQ créées. Cet enregistrement peut être utilisé pour:

- Objectifs de reprise
- Maintenance
- Déploiement d'applications IBM MQ

Vous pouvez enregistrer les définitions d'application IBM MQ de l'une des deux manières suivantes:

1. Création de programmes CL pour générer vos définitions IBM MQ pour le serveur.
2. Création de fichiers texte MQSC en tant que membres SRC pour générer vos définitions IBM MQ à l'aide du langage de commande IBM MQ multiplateforme.

Pour plus de détails sur la définition des objets de file d'attente, voir «Administration d' IBM MQ à l'aide de commandes MQSC», à la page 12 et «Utilisation des formats de commande programmables IBM MQ», à la page 28.

### Référence associée

Référence des commandes CL IBM MQ for IBM i

## Avant de commencer à utiliser IBM MQ for IBM i à l'aide de commandes CL

Utilisez ces informations pour démarrer le sous-système IBM MQ et créer un gestionnaire de files d'attente local.

### Avant de commencer

Vérifiez que le sous-système IBM MQ est en cours d'exécution (à l'aide de la commande STRSBS QMQM/ QMQM ) et que la file d'attente de travaux associée à ce sous-système n'est pas suspendue. Par défaut, le sous-système IBM MQ et la file d'attente de travaux sont tous deux nommés QMQM dans la bibliothèque QMQM.

### Pourquoi et quand exécuter cette tâche

Utilisation de la ligne de commande IBM i pour démarrer un gestionnaire de files d'attente

### Procédure

1. Créez un gestionnaire de files d'attente local en émettant la commande CRTMQM à partir d'une ligne de commande IBM i .

Lorsque vous créez un gestionnaire de files d'attente, vous avez la possibilité de faire de ce gestionnaire de files d'attente le gestionnaire de files d'attente par défaut. Le gestionnaire de files d'attente par défaut (dont un seul peut exister) est le gestionnaire de files d'attente auquel s'applique une commande CL, si le paramètre de nom de gestionnaire de files d'attente (MQMNAME) est omis.

2. Démarrez un gestionnaire de files d'attente local en exécutant la commande STRMQM à partir d'une ligne de commande IBM i .

Si le démarrage du gestionnaire de files d'attente prend plus de quelques secondes, IBM MQ affiche des messages d'état détaillant par intermittence la progression du démarrage. Pour plus d'informations sur ces messages, voir [Messages et codes anomalie](#).

### Que faire ensuite

Vous pouvez arrêter un gestionnaire de files d'attente en émettant la commande ENDMQM à partir de la ligne de commande IBM i et en contrôlant un gestionnaire de files d'attente en émettant d'autres commandes IBM MQ à partir d'une ligne de commande IBM i .

Les gestionnaires de files d'attente éloignées ne peuvent pas être démarrés à distance mais doivent être créés et démarrés sur leurs systèmes par des opérateurs locaux. Une exception à cette règle est l'existence de fonctions d'exploitation à distance (en dehors de IBM MQ for IBM i) permettant d'activer de telles opérations.

L'administrateur de file d'attente locale ne peut pas arrêter un gestionnaire de files d'attente éloignées.



**Remarque :** Dans le cadre de la mise au repos d'un système IBM MQ , vous devez mettre au repos les gestionnaires de files d'attente actifs. Ceci est décrit dans [«mise au repos IBM MQ for IBM i»](#), à la page 465.

## **Création d'objets IBM MQ for IBM i**

Utilisez ces informations pour comprendre les méthodes de création d'objets IBM MQ pour IBM i.

### **Avant de commencer**

Les tâches suivantes suggèrent différentes manières d'utiliser IBM MQ for IBM i à partir de la ligne de commande.

### **Pourquoi et quand exécuter cette tâche**

Il existe deux méthodes en ligne pour créer des objets IBM MQ , à savoir:

#### **Procédure**

1. A l'aide d'une commande Create, par exemple: la commande **Create MQM Queue : CRTMQMQ**
2. A l'aide d'une commande Work with MQM object, suivie de F6, par exemple: La commande **Work with MQM Queues : WRKMQM**

#### **Que faire ensuite**

Pour obtenir la liste de toutes les commandes, voir [Commandes CLIBM MQ for IBM i](#).

**Remarque :** Toutes les commandes MQM peuvent être soumises à partir du menu Commandes gestionnaire de files d'attente de messages. Pour afficher ce menu, entrez GO CMDMQM sur la ligne de commande et appuyez sur la touche Enter .

Le système affiche automatiquement le panneau d'invite lorsque vous sélectionnez une commande dans ce menu. Pour afficher le panneau d'invite d'une commande que vous avez saisie directement sur la ligne de commande, appuyez sur F4 avant d'appuyer sur la touche Enter .

### ***Création d'une file d'attente locale à l'aide de la commande CRTMQMQ***

#### **Procédure**

1. Entrez CHGMQM sur la ligne de commande et appuyez sur la touche F4 .
2. Dans le panneau **Créer une file d'attente MQM**, entrez le nom de la file d'attente à créer dans la zone Queue name . Pour indiquer un nom à casse mixte, placez le nom entre apostrophes.
3. Entrez \*LCL dans la zone Queue type .
4. Indiquez un nom de gestionnaire de files d'attente, sauf si vous utilisez le gestionnaire de files d'attente par défaut, puis appuyez sur la touche Enter . Vous pouvez remplacer n'importe quelle valeur par une nouvelle valeur. Faites défiler vers l'avant pour afficher d'autres zones. Les options utilisées pour les clusters se trouvent à la fin de la liste des options.
5. Lorsque vous avez modifié des valeurs, appuyez sur la touche Enter pour créer la file d'attente.

### ***Création d'une file d'attente locale à l'aide de la commande WRKMQM***

#### **Procédure**

1. Entrez WRKMQM sur la ligne de commande.
2. Entrez le nom d'un gestionnaire de files d'attente.
3. Si vous souhaitez afficher le panneau d'invite, appuyez sur F4. Le panneau d'invite permet de réduire le nombre de files d'attente affichées en indiquant un nom de file d'attente générique ou un type de file d'attente.

4. Appuyez sur **Enter** et le panneau **Work with MQM Queues** s'affiche. Vous pouvez remplacer n'importe quelle valeur par une nouvelle valeur. Faites défiler vers l'avant pour afficher d'autres zones. Les options utilisées pour les clusters se trouvent à la fin de la liste des options.
5. Appuyez sur **F6** pour créer une nouvelle file d'attente. Vous accédez alors au panneau **CRTMQMQ**. Pour savoir comment créer la file d'attente, voir «Création d'une file d'attente locale à l'aide de la commande CRTMQMQ», à la page 393. Une fois que vous avez créé la file d'attente, le **panneau Work with MQM Queues** s'affiche à nouveau. La nouvelle file d'attente est ajoutée à la liste lorsque vous appuyez sur **F5=Refresh**.

## Modification des attributs du gestionnaire de files d'attente

### Pourquoi et quand exécuter cette tâche

Pour modifier les attributs du gestionnaire de files d'attente spécifié dans la commande **CHGMQM**, indiquez les attributs et les valeurs à modifier. Par exemple, utilisez les options suivantes pour modifier les attributs de `jupiter.queue.manager`:

### Procédure

Entrez **CHGMQM** sur la ligne de commande et appuyez sur la touche **F4**.

### Résultats

La commande modifie la file d'attente de rebut utilisée et active les événements d'interdiction.

## IBM i Utilisation des files d'attente locales sous IBM i

Cette section contient des exemples de certaines des commandes que vous pouvez utiliser pour gérer les files d'attente locales. Toutes les commandes affichées sont également disponibles à l'aide des options du **panneau de commande WRKMQM**.

### Définition d'une file d'attente locale

Pour une application, le gestionnaire de files d'attente local est le gestionnaire de files d'attente auquel l'application est connectée. Les files d'attente gérées par le gestionnaire de files d'attente local sont dites locales pour ce gestionnaire de files d'attente.

Utilisez la commande **CRTMQMQ QTYPE \*LCL** pour créer une définition de file d'attente locale et pour créer la structure de données appelée file d'attente. Vous pouvez également modifier les caractéristiques de la file d'attente à partir de celles de la file d'attente locale par défaut.

Dans cet exemple, la file d'attente que nous définissons, `orange.local.queue`, possède les caractéristiques suivantes:

- Il est activé pour les extractions, désactivé pour les insertions et fonctionne sur la base du premier entré, premier sorti (FIFO).
- Il s'agit d'une file d'attente *ordinaire*, c'est-à-dire qu'il ne s'agit pas d'une file d'attente d'initialisation ou de transmission et qu'elle ne génère pas de messages de déclenchement.
- La longueur maximale de la file d'attente est de 1000 messages ; la longueur maximale est de 2000 octets.

La commande suivante effectue cette opération sur le gestionnaire de files d'attente par défaut:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL)
TEXT('Queue for messages from other systems')
PUTENBL(*NO)
GETENBL(*YES)
TRGENBL(*NO)
MSGDLYSEQ(*FIFO)
MAXDEPTH(1000)
```

### Remarque :

1. USAGE \*NORMAL indique que cette file d'attente n'est pas une file d'attente de transmission.
2. Si vous disposez déjà d'une file d'attente locale portant le nom `orange.local.queue` sur le même gestionnaire de files d'attente, cette commande échoue. Utilisez l'attribut REPLACE \*YES si vous souhaitez remplacer la définition existante d'une file d'attente, mais voir aussi [«Modification des attributs de file d'attente locale»](#), à la page 396.

## Définition d'une file d'attente de rebut

Chaque gestionnaire de files d'attente doit disposer d'une file d'attente locale à utiliser comme file d'attente de rebut pour que les messages qui ne peuvent pas être distribués à leur destination correcte puissent être stockés en vue d'une extraction ultérieure. Vous devez indiquer explicitement au gestionnaire de files d'attente la file d'attente de rebut. Pour ce faire, vous pouvez spécifier une file d'attente de rebut dans la commande **CRTMQM** ou utiliser la commande **CHGMQM** pour en spécifier une ultérieurement. Vous devez également définir la file d'attente de rebut avant de pouvoir l'utiliser.

Un exemple de file d'attente de rebut appelé `SYSTEM.DEAD.LETTER.QUEUE` est fourni avec le produit. Cette file d'attente est automatiquement créée lorsque vous créez le gestionnaire de files d'attente. Vous pouvez modifier cette définition si nécessaire. Il n'est pas nécessaire de le renommer, même si vous pouvez le faire si vous le souhaitez.

Une file d'attente de rebut n'a pas d'exigences particulières si ce n'est que:

- Il doit s'agir d'une file d'attente locale.
- Son attribut MAXMSGL (longueur maximale de message) doit permettre à la file d'attente de prendre en charge les messages les plus volumineux que le gestionnaire de files d'attente doit traiter **plus** la taille de l'en-tête de rebut (MQDLH).

IBM MQ fournit un gestionnaire de files d'attente de rebut qui permet d'indiquer comment les messages trouvés dans une file d'attente de rebut doivent être traités ou supprimés. Pour plus d'informations, reportez-vous à la section [«Appel du gestionnaire de files d'attente de rebut sous IBM i»](#), à la page 174.

## Affichage des attributs d'objet par défaut

Lorsque vous définissez un objet IBM MQ, il prend tous les attributs que vous ne spécifiez pas à partir de l'objet par défaut. Par exemple, lorsque vous définissez une file d'attente locale, celle-ci hérite de tous les attributs que vous omettez dans la définition de la file d'attente locale par défaut, appelée `SYSTEM.DEFAULT.LOCAL.QUEUE`. Pour voir exactement quels sont ces attributs, utilisez la commande suivante:

```
DSPMQMQ QNAME(SYSTEM.DEFAULT.LOCAL.QUEUE) MQMNAME(MYQUEUEMANAGER)
```

## Copie d'une définition de file d'attente locale

Vous pouvez copier une définition de file d'attente à l'aide de la commande `CPYMQMQ`. Exemple :

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Cette commande crée une file d'attente avec les mêmes attributs que notre file d'attente d'origine `orange.local.queue`, plutôt que ceux de la file d'attente locale par défaut du système.

Vous pouvez également utiliser la commande **CPYMQMQ** pour copier une définition de file d'attente, mais en remplaçant une ou plusieurs modifications apportées aux attributs de l'original. Exemple :

```
CPYMQMQ FROMQ('orange.local.queue') TOQ('third.queue') MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(1024)
```

Cette commande copie les attributs de la file d'attente `orange.local.queue` dans la file d'attente `third.queue`, mais indique que la longueur maximale des messages dans la nouvelle file d'attente doit être de 1024 octets au lieu de 2000.

**Remarque :** Lorsque vous utilisez la commande **CPYMQMQ**, vous copiez uniquement les attributs de file d'attente et non les messages de la file d'attente.

## Modification des attributs de file d'attente locale

Vous pouvez modifier les attributs de file d'attente de deux manières, à l'aide de la commande **CHGMQMQ** ou de la commande **CPYMQMQ** avec l'attribut `REPLACE *YES`. Dans «[Définition d'une file d'attente locale](#)», à la page 394, vous avez défini la file d'attente `orange.local.queue`. Si, par exemple, vous devez augmenter la longueur maximale des messages dans cette file d'attente à 10 000 octets.

- A l'aide de la commande **CHGMQMQ** :

```
CHGMQMQ QNAME('orange.local.queue') MQMNAME(MYQUEUEMANAGER) MAXMSGLEN(10000)
```

Cette commande modifie un attribut unique, celui de la longueur maximale de message ; tous les autres attributs restent identiques.

- A l'aide de la commande **CRTMQMQ** avec l'option `REPLACE *YES`, par exemple:

```
CRTMQMQ QNAME('orange.local.queue') QTYPE(*LCL) MQMNAME(MYQUEUEMANAGER)
MAXMSGLEN(10000) REPLACE(*YES)
```

Cette commande modifie non seulement la longueur maximale des messages, mais également tous les autres attributs, auxquels sont attribuées leurs valeurs par défaut. La file d'attente est maintenant mise en file d'attente activée alors qu'auparavant elle était mise en file d'attente interdite. L'insertion activée est la valeur par défaut, comme indiqué par la file d'attente `SYSTEM.DEFAULT.LOCAL.QUEUE`, sauf si vous l'avez modifiée.

Si vous **réduisez** la longueur maximale des messages dans une file d'attente existante, les messages existants ne sont pas affectés. Toutefois, tout nouveau message doit répondre aux nouveaux critères.

## Effacement d'une file d'attente locale

Pour supprimer tous les messages d'une file d'attente locale appelée `magenta.queue`, utilisez la commande suivante:

```
CLRMQMQ QNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Vous ne pouvez pas effacer une file d'attente si:

- Des messages non validés ont été insérés dans la file d'attente lors d'un point de synchronisation.
- La file d'attente est ouverte par une application.

## Suppression d'une file d'attente locale

Utilisez la commande **DLTMQMQ** pour supprimer une file d'attente locale.

Une file d'attente ne peut pas être supprimée si elle contient des messages non validés ou si elle est en cours d'utilisation.

## Activation des files d'attente volumineuses

IBM MQ prend en charge les files d'attente supérieures à 2 Go. Consultez la documentation de votre système d'exploitation pour savoir comment activer IBM i pour la prise en charge des fichiers volumineux.

Les informations sur le produit IBM i sont disponibles dans [IBM Documentation](#).

Certains utilitaires peuvent ne pas être en mesure de gérer les fichiers de plus de 2 Go. Avant d'activer la prise en charge des fichiers volumineux, consultez la documentation de votre système d'exploitation pour plus d'informations sur les restrictions liées à cette prise en charge.

## IBM i Utilisation des files d'attente alias sous IBM i

Cette section contient des exemples de certaines des commandes que vous pouvez utiliser pour gérer les files d'attente alias. Toutes les commandes affichées sont également disponibles à l'aide des options du **panneau de commande WRKMQMQ**.

Une file d'attente alias (parfois appelée alias de file d'attente) fournit une méthode de redirection des appels MQI. Une file d'attente alias n'est pas une file d'attente réelle, mais une définition qui se résout en file d'attente réelle. La définition de file d'attente alias contient un nom de file d'attente cible, qui est spécifié par l'attribut TGTQNAME .

Lorsqu'une application spécifie une file d'attente alias dans un appel MQI, le gestionnaire de files d'attente résout le nom de la file d'attente réelle lors de l'exécution.

Par exemple, une application a été développée pour placer des messages dans une file d'attente appelée `my.alias.queue`. Il indique le nom de cette file d'attente lorsqu'il émet une demande **MQOPEN** et, indirectement, s'il insère un message dans cette file d'attente. L'application ne sait pas que la file d'attente est une file d'attente d'alias. Pour chaque appel MQI utilisant cet alias, le gestionnaire de files d'attente résout le nom de la file d'attente réelle, qui peut être une file d'attente locale ou une file d'attente éloignée définie sur ce gestionnaire de files d'attente.

En modifiant la valeur de l'attribut TGTQNAME , vous pouvez rediriger les appels MQI vers une autre file d'attente, éventuellement sur un autre gestionnaire de files d'attente. Cela est utile pour la maintenance, la migration et l'équilibrage de charge.

## Définition d'une file d'attente alias

La commande suivante crée une file d'attente alias:

```
CRTMQMQ QNAME('my.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
MQMNAME(MYQUEUEMANAGER)
```

Cette commande redirige les appels MQI qui spécifient `my.alias.queue` vers la file d'attente `yellow.queue`. La commande ne crée pas la file d'attente cible ; les appels MQI échouent si la file d'attente `yellow.queue` n'existe pas lors de l'exécution.

Si vous modifiez la définition d'alias, vous pouvez rediriger les appels MQI vers une autre file d'attente. Exemple :

```
CHGMQM QNAME('my.alias.queue') TGTQNAME('magenta.queue') MQMNAME(MYQUEUEMANAGER)
```

Cette commande redirige les appels MQI vers une autre file d'attente, `magenta.queue`.

Vous pouvez également utiliser des files d'attente alias pour qu'une seule file d'attente (la file d'attente cible) ait des attributs différents pour différentes applications. Pour ce faire, vous définissez deux alias, un pour chaque application. Supposons qu'il existe deux applications:

- L'application ALPHA peut insérer des messages dans `yellow.queue`, mais elle n'est pas autorisée à en extraire des messages.
- L'application BETA peut extraire des messages de `yellow.queue`, mais elle n'est pas autorisée à y placer des messages.

Pour ce faire, utilisez les commandes suivantes:

```
/* This alias is put enabled and get disabled for application ALPHA */
CRTMQMQ QNAME('alphas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*YES) GETENBL(*NO) MQMNAME(MYQUEUEMANAGER)

/* This alias is put disabled and get enabled for application BETA */
CRTMQMQ QNAME('betas.alias.queue') QTYPE(*ALS) TGTQNAME('yellow.queue')
PUTENBL(*NO) GETENBL(*YES) MQMNAME(MYQUEUEMANAGER)
```

ALPHA utilise le nom de file d'attente `alphas.alias.queue` dans ses appels MQI ; BETA utilise le nom de file d'attente `betas.alias.queue`. Ils accèdent tous deux à la même file d'attente, mais de différentes manières.

Vous pouvez utiliser l'attribut `REPLACE *YES` lorsque vous définissez des files d'attente alias, de la même manière que vous utilisez ces attributs avec des files d'attente locales.

## Utilisation d'autres commandes avec des files d'attente alias

Vous pouvez utiliser les commandes appropriées pour afficher ou modifier les attributs de file d'attente alias. Exemple :

```
* Display the alias queue's attributes */
DSPMQMQ QNAME('alphas.alias.queue') MQMNAME(MYQUEUEMANAGER)

/* ALTER the base queue name, to which the alias resolves. */
/* FORCE = Force the change even if the queue is open. */
CHMQMQ QNAME('alphas.alias.queue') TGTQNAME('orange.local.queue') FORCE(*YES)
MQMNAME(MYQUEUEMANAGER)
```



## Utilisation des files d'attente modèles sous IBM i

Cette section contient des exemples de certaines des commandes que vous pouvez utiliser pour gérer les files d'attente modèles. Toutes les commandes affichées sont également disponibles à l'aide des options du **panneau de commande WRKMQM**.

Un gestionnaire de files d'attente crée une file d'attente dynamique s'il reçoit un appel MQI d'une application spécifiant un nom de file d'attente qui a été défini en tant que file d'attente modèle. Le nom de la nouvelle file d'attente dynamique est généré par le gestionnaire de files d'attente lors de la création de la file d'attente. Une file d'attente modèle est un modèle qui spécifie les attributs des files d'attente dynamiques créées à partir de celle-ci.

Les files d'attente modèles constituent une méthode pratique permettant aux applications de créer des files d'attente au fur et à mesure qu'elles sont requises.

## Définition d'une file d'attente modèle

Vous définissez une file d'attente modèle avec un ensemble d'attributs de la même manière que vous définissez une file d'attente locale. Les files d'attente modèles et les files d'attente locales ont le même ensemble d'attributs, sauf que dans les files d'attente modèles, vous pouvez indiquer si les files d'attente dynamiques créées sont temporaires ou permanentes. (Les files d'attente permanentes sont gérées lors des redémarrages du gestionnaire de files d'attente, alors que les files d'attente temporaires ne le sont pas). Exemple :

```
CRTMQMQ QNAME('green.model.queue') QTYPE(*MDL) DFNTYPE(*PERMDYN)
```

Cette commande crée une définition de file d'attente modèle. A partir de l'attribut `DFNTYPE`, les files d'attente réelles créées à partir de ce modèle sont des files d'attente dynamiques permanentes.

Les attributs non spécifiés sont automatiquement copiés à partir de la file d'attente par défaut `SYSYSTEM.DEFAULT.MODEL.QUEUE`.

Vous pouvez utiliser l'attribut `REPLACE *YES` lorsque vous définissez des files d'attente modèles, de la même manière que vous les utilisez avec des files d'attente locales.

## Utilisation d'autres commandes avec des files d'attente modèles

Vous pouvez utiliser les commandes appropriées pour afficher ou modifier les attributs d'une file d'attente modèle. Exemple :

```
/* Display the model queue's attributes */
DSPMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('green.model.queue')

/* ALTER the model queue to enable puts on any */
/* dynamic queue created from this model. */
CHGMQM MQMNAME(MYQUEUEMANAGER) QNAME('blue.model.queue') PUTENBL(*YES)
```

IBM i

## Utilisation du déclenchement sous IBM i

Utilisez ces informations pour en savoir plus sur le déclenchement et les définitions de processus.

IBM MQ fournit une fonction permettant de démarrer automatiquement une application lorsque certaines conditions d'une file d'attente sont remplies. Par exemple, lorsque le nombre de messages dans une file d'attente atteint un nombre spécifié. Cette fonction est appelée *déclenchement* et est décrite en détail dans [Déclenchement des canaux](#).

### Qu'est-ce qui se déclenche?

Le gestionnaire de files d'attente définit certaines conditions comme constituant des événements déclencheurs. Si le déclenchement est activé pour une file d'attente et qu'un événement déclencheur se produit, le gestionnaire de files d'attente envoie un message de déclenchement à une file d'attente appelée file d'attente d'initialisation. La présence du message de déclenchement dans la file d'attente d'initialisation indique qu'un événement déclencheur s'est produit.

Les messages de déclenchement générés par le gestionnaire de files d'attente ne sont pas persistants. Cela a pour effet de réduire la consignation (améliorant ainsi les performances) et de réduire les doublons lors du redémarrage, ce qui améliore le temps de redémarrage.

### Qu'est-ce que le moniteur de déclenchement?

Le programme qui traite la file d'attente d'initialisation est appelé une application de moniteur de déclenchement et sa fonction est de lire le message de déclenchement et de prendre les mesures appropriées, en fonction des informations contenues dans le message de déclenchement. Normalement, cette action consiste à démarrer une autre application pour traiter la file d'attente à l'origine de la génération du message de déclenchement. Du point de vue du gestionnaire de files d'attente, il n'y a rien de spécial à propos de l'application du moniteur de déclenchement: c'est une autre application qui lit les messages d'une file d'attente (la file d'attente d'initialisation).

### Modification des attributs de soumission de travaux du moniteur de déclenchement

Le moniteur de déclenchement fourni en tant que commande `STRMQMTRM` soumet un travail pour chaque message de déclenchement à l'aide de la description de travail par défaut du système, `QDFTJOB`. Cela comporte des limitations dans la mesure où les travaux soumis sont toujours appelés `QDFTJOB` et possèdent les attributs de la description de travail par défaut, y compris la liste des bibliothèques, `*SYSVAL`. IBM MQ fournit une méthode pour remplacer ces attributs. Par exemple, il est possible de personnaliser les travaux soumis pour qu'ils aient des noms de travail plus significatifs, comme suit:

1. Dans la description de travail, indiquez la description de votre choix, par exemple les valeurs de consignation.
2. Indiquez les données d'environnement de la définition de processus utilisée dans le processus de déclenchement:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)')
```

Le moniteur de déclenchement exécute un SBMJOB à l'aide de la description spécifiée.

Il est possible de remplacer d'autres attributs du SBMJOB en spécifiant le mot clé et la valeur appropriés dans les données d'environnement de la définition de processus. La seule exception à cette règle est le mot clé CMD car cet attribut est rempli par le moniteur de déclenchement. Voici un exemple de commande permettant de spécifier les données d'environnement de la définition de processus dans laquelle le nom et la description du travail doivent être modifiés:

```
CHGMQMPRC PRCNAME(MY_PROCESS) MQMNAME(MHA3) ENVDATA ('JOB(MYLIB/TRIGJOB)
JOB(TRIGGER)')
```

## Définition d'une file d'attente d'application pour le déclenchement

Une file d'attente d'application est une file d'attente locale utilisée par les applications pour la messagerie, via l'interface MQI. Le déclenchement requiert la définition d'un certain nombre d'attributs de file d'attente dans la file d'attente d'application. Le déclenchement est activé par l'attribut TRGENBL .

Dans cet exemple, un événement déclencheur doit être généré lorsqu'il y a 100 messages de priorité 5 ou supérieure dans la file d'attente locale `motor.insurance.queue`, comme suit:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.insurance.queue') QTYPE(*LCL)
PRCNAME('motor.insurance.quote.process') MAXMSGLEN(2000)
DFTMSGPST(*YES) INITQNAME('motor.ins.init.queue')
TRGENBL(*YES) TRGTYPE(*DEPTH) TRGDEPTH(100) TRGMSGPTY(5)
```

où les paramètres sont :

### **MQMNAME(MYQUEUEMANAGER)**

Nom du gestionnaire de files d'attente.

### **QNAME('motor.insurance.queue')**

Nom de la file d'attente d'application en cours de définition.

### **PRCNAME('motor.insurance.quote.process')**

Nom de l'application devant être démarrée par un programme de moniteur de déclenchement.

### **MAXMSGLEN(2000)**

Longueur maximale des messages dans la file d'attente.

### **DFTMSGPST(\*YES)**

Les messages de cette file d'attente sont persistants par défaut.

### **INITQNAME('motor.ins.init.queue')**

Nom de la file d'attente d'initialisation dans laquelle le gestionnaire de files d'attente doit insérer le message de déclenchement.

### **TRGENBL(\*YES)**

Valeur de l'attribut de déclencheur.

### **TRGTYPE(\*DEPTH)**

Un événement déclencheur est généré lorsque le nombre de messages de la priorité requise (**TRGMSGPTY**) atteint le nombre spécifié dans **TRGDEPTH**.

### **TRGDEPTH(100)**

Nombre de messages requis pour générer un événement déclencheur.



## TRGMSGPTY(5)

Priorité des messages qui doivent être comptés par le gestionnaire de files d'attente pour déterminer s'il convient de générer un événement déclencheur. Seuls les messages de priorité 5 ou supérieure sont comptés.

## Définition d'une file d'attente d'initialisation

Lorsqu'un événement déclencheur se produit, le gestionnaire de files d'attente place un message de déclenchement dans la file d'attente d'initialisation spécifiée dans la définition de file d'attente d'application. Les files d'attente d'initialisation n'ont pas de paramètres spéciaux, mais vous pouvez utiliser la définition suivante de la file d'attente locale `motor.ins.init.queue` pour obtenir de l'aide:

```
CRTMQMQ MQMNAME(MYQUEUEMANAGER) QNAME('motor.ins.init.queue') QTYPE(*LCL)
GETENBL(*YES) SHARE(*NO) TRGTYPE(*NONE)
MAXMSGL(2000)
MAXDEPTH(1000)
```

## Création d'une définition de processus

La commande **CRTMQMPRC** permet de créer une définition de processus. Une définition de processus associe une file d'attente d'application à l'application qui doit traiter les messages de la file d'attente. Cette opération est effectuée via l'attribut `PRCNAME` de la file d'attente d'application `motor.insurance.quote.queue`. La commande suivante crée le processus requis, `motor.insurance.quote.process`, identifié dans cet exemple:

```
CRTMQMPRC MQMNAME(MYQUEUEMANAGER) PRCNAME('motor.insurance.quote.process')
TEXT('Insurance request message processing')
APPTYPE(*OS400) APPID(MQTEST/TESTPROG)
USRDATA('open, close, 235')
```

où les paramètres sont :

### **MQMNAME(MYQUEUEMANAGER)**

Nom du gestionnaire de files d'attente.

### **PRCNAME('motor.insurance.quote.process')**

Nom de la définition de processus.

### **TEXT('Insurance request message processing')**

Description du programme d'application auquel cette définition est liée. Ce texte s'affiche lorsque vous utilisez la commande **DSPMQMPRC**. Cela peut vous aider à identifier ce que le processus fait. Si vous utilisez des espaces dans la chaîne, vous devez placer la chaîne entre apostrophes.

### **APPTYPE(\*OS400)**

Type d'application à démarrer.

### **APPID(MQTEST/TESTPROG)**

Nom du fichier exécutable de l'application, spécifié en tant que nom de fichier qualifié complet.

### **USRDATA('open, close, 235')**

Données définies par l'utilisateur, qui peuvent être utilisées par l'application.

## Affichage de votre définition de processus

Utilisez la commande **DSPMQMPRC** pour examiner les résultats de votre définition. Exemple :

```
MQMNAME(MYQUEUEMANAGER) DSPMQMPRC('motor.insurance.quote.process')
```

Vous pouvez également utiliser la commande **CHGMQMPRC** pour modifier une définition de processus existante et la commande **DLTMQMPRC** pour supprimer une définition de processus.

Cet exemple de codage montre comment configurer deux systèmes IBM MQ for IBM i, à l'aide de commandes CL, pour qu'ils puissent communiquer entre eux.

Les systèmes sont appelés SYSTEMA et SYSTEMB et le protocole de communication utilisé est TCP/IP.

Effectuez la procédure suivante :

1. Créez un gestionnaire de files d'attente sous SYSTEMA, en l'appelant QMGRA1.

```
CRTMQM  MQMNAME(QMGRA1) TEXT('System A - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

2. Démarrez ce gestionnaire de files d'attente.

```
STRMQM  MQMNAME(QMGRA1)
```

3. Définissez les objets IBM MQ sur SYSTEMA dont vous avez besoin pour envoyer des messages à un gestionnaire de files d'attente sur SYSTEMB.

```
/* Transmission queue */
CRTMQMQ  QNAME(XMITQ.TO.QMGRB1) QTYPE(*LCL) +
MQMNAME(QMGRA1) TEXT('Transmission Queue +
to QMGRB1') MAXDEPTH(5000) USAGE(*TMQ)

/* Remote queue that points to a queue called TARGETB */
/* TARGETB belongs to queue manager QMGRB1 on SYSTEMB */
CRTMQMQ  QNAME(TARGETB.ON.QMGRB1) QTYPE(*RMT) +
MQMNAME(QMGRA1) TEXT('Remote Q pointing +
at Q TARGETB on QMGRB1 on Remote System +
SYSTEMB') RMTQNAME(TARGETB) +
RMTMQMNAME(QMGRB1) TMQNAME(XMITQ.TO.QMGRB1)

/* TCP/IP sender channel to send messages to the queue manager on SYSTEMB*/
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*SDR) +
MQMNAME(QMGRA1) TRPTYPE(*TCP) +
TEXT('Sender Channel From QMGRA1 on +
SYSTEMA to QMGRB1 on SYSTEMB') +
CONNNAME(SYSTEMB) TMQNAME(XMITQ.TO.QMGRB1)
```

4. Créez un gestionnaire de files d'attente sous SYSTEMB, en l'appelant QMGRB1.

```
CRTMQM  MQMNAME(QMGRB1) TEXT('System B - Queue +
Manager 1') UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
```

5. Démarrez le gestionnaire de files d'attente sous SYSTEMB.

```
STRMQM  MQMNAME(QMGRB1)
```

6. Définissez les objets IBM MQ dont vous avez besoin pour recevoir des messages du gestionnaire de files d'attente sous SYSTEMA.

```
/* Local queue to receive messages on */
CRTMQMQ  QNAME(TARGETB) QTYPE(*LCL) MQMNAME(QMGRB1) +
TEXT('Sample Local Queue for QMGRB1')

/* Receiver channel of the same name as the sender channel on SYSTEMA */
CRTMQMCHL CHLNAME(QMGRA1.TO.QMGRB1) CHLTYPE(*RCVR) +
MQMNAME(QMGRB1) TRPTYPE(*TCP) +
TEXT('Receiver Channel from QMGRA1 to +
QMGRB1')
```

- Enfin, démarrez un programme d'écoute TCP/IP sur SYSTEMB afin que le canal puisse être démarré. Cet exemple utilise le port par défaut 1414.

```
STRMQLSR MQMNAME(QMGRB1)
```

Vous êtes maintenant prêt à envoyer des messages de test entre SYSTEMA et SYSTEMB. A l'aide de l'un des exemples fournis, placez une série de messages dans votre file d'attente éloignée sur SYSTEMA.

Démarrez le canal sur SYSTEMA, soit à l'aide de la commande **STRMQMCHL**, soit à l'aide de la commande **WRKMQMCHL** et en entrant une demande de démarrage (option 14) sur le canal émetteur.

Le canal doit passer à l'état RUNNING et les messages sont envoyés à la file d'attente TARGETB sur SYSTEMB.

Vérifiez vos messages en exécutant la commande suivante:

```
WRKMQMMSG QNAME(TARGETB) MQMNAME(QMGRB1).
```

## Exemples de définitions de ressource sous IBM i

Cet exemple contient l'exemple de programme CL AMQSAMP4 IBM i.

```
/*
/*
/* *****
/*
/* Program name: AMQSAMP4
/*
/*
/* Description: Sample CL program defining MQM queues
/* to use with the sample programs
/* Can be run, with changes as needed, after
/* starting the MQM
/*
/*
/* <N_OCO_COPYRIGHT>
/* Licensed Materials - Property of IBM
/*
/* 63H9336
/* (c) Copyright IBM Corp. 1993, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* <NOC_COPYRIGHT>
/*
/* *****
/*
/* Function:
/*
/*
/* AMQSAMP4 is a sample CL program to create or reset the
/* MQI resources to use with the sample programs.
/*
/* This program, or a similar one, can be run when the MQM
/* is started - it creates the objects if missing, or resets
/* their attributes to the prescribed values.
/*
/*
/*
/*
/* Exceptions signaled: none
/* Exceptions monitored: none
/*
/*
/* AMQSAMP4 takes a single parameter, the Queue Manager name
/*
/* *****
/*SYS/PGM PARM(&QMGRNAME)
/*
/* *****
/* Queue Manager Name Parameter
/*
/* *****
/*SYS/DCL VAR(&QMGRNAME) TYPE(*CHAR)
/*
/* *****
/*
/* EXAMPLES OF DIFFERENT QUEUE TYPES
/*
```

```

/*                                                                    */
/* Create local, alias and remote queues                               */
/*                                                                    */
/* Uses system defaults for most attributes                           */
/*                                                                    */
/*****
/* Create a local queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.LOCAL') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Sample local queue') /* description */+
SHARE(*YES) /* Shareable */+
DFTMSGPST(*YES) /* Persistent messages OK */

/* Create an alias queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.ALIAS') +
MQMNAME(&QMGRNAME) +
QTYPE(*ALS) REPLACE(*YES) +
+
TEXT('Sample alias queue') +
DFTMSGPST(*YES) /* Persistent messages OK */+
TGTONAME('SYSTEM.SAMPLE.LOCAL')

/* Create a remote queue - in this case, an indirect reference */
/* is made to the sample local queue on OTHER queue manager */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REMOTE') +
MQMNAME(&QMGRNAME) +
QTYPE(*RMT) REPLACE(*YES) +
+
TEXT('Sample remote queue')/* description */+
DFTMSGPST(*YES) /* Persistent messages OK */+
RMTQNAME('SYSTEM.SAMPLE.LOCAL') +
RMTMQMNAME(OTHER) /* Queue is on OTHER */

/* Create a transmission queue for messages to queues at OTHER */
/* By default, use remote node name */
CRTMQMQ QNAME('OTHER') /* transmission queue name */+
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
TEXT('Transmission queue to OTHER') +
USAGE(*TMQ) /* transmission queue */

/*****
/* SPECIFIC QUEUES AND PROCESS USED BY SAMPLE PROGRAMS */
/*                                                                    */
/* Create local queues used by sample programs                       */
/* Create MQI process associated with sample initiation queue       */
/*                                                                    */
/*****
/* General reply queue */
CRTMQMQ QNAME('SYSTEM.SAMPLE.REPLY') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('General reply queue') +
DFTMSGPST(*NO) /* Not Persistent */

/* Queue used by AMQSINQ4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.INQ') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSINQ4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO) /* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSSET4 */
CRTMQMQ QNAME('SYSTEM.SAMPLE.SET') +
MQMNAME(&QMGRNAME) +
QTYPE(*LCL) REPLACE(*YES) +
+
TEXT('Queue for AMQSSET4') +
SHARE(*YES) /* Shareable */+
DFTMSGPST(*NO)/* Not Persistent */+
+
TRGENBL(*YES) /* Trigger control on */+

```

```

TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.SETPROCESS')      +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Queue used by AMQSECH4                               */
CRTMQMQ  QNAME('SYSTEM.SAMPLE.ECHO')      +
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                +
+
TEXT('Queue for AMQSECH4')                +
SHARE(*YES) /* Shareable */ +
DFTMSGPST(*NO)/* Not Persistent */ +
+
TRGENBL(*YES) /* Trigger control on */ +
TRGTYPE(*FIRST)/* Trigger on first message*/+
PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS')      +
INITQNAME('SYSTEM.SAMPLE.TRIGGER')

/* Initiation Queue used by AMQSTRG4, sample trigger process */
CRTMQMQ  QNAME('SYSTEM.SAMPLE.TRIGGER') +
MQMNAME(&QMGRNAME)                        +
QTYPE(*LCL) REPLACE(*YES)                +
TEXT('Trigger queue for sample programs')

/* MQI Processes associated with triggered sample programs */
/*                               */
/***** Note - there are versions of the triggered samples *****/
/***** in different languages - set APPID for these *****/
/***** process to the variation you want to trigger *****/
/*                               */
CRTMQMPC PRCNAME('SYSTEM.SAMPLE.INQPROCESS') +
MQMNAME(&QMGRNAME)                        +
REPLACE(*YES)                             +
+
TEXT('Trigger process for AMQSINQ4')      +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSINQ4') /* C */ +
/* APPID('QMOM/AMQ0INQ4') /* COBOL */ +
/* APPID('QMOM/AMQ3INQ4') /* RPG - ILE */

CRTMQMPC PRCNAME('SYSTEM.SAMPLE.SETPROCESS') +
MQMNAME(&QMGRNAME)                        +
REPLACE(*YES)                             +
+
TEXT('Trigger process for AMQSSET4')      +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSSET4') /* C */ +
/* APPID('QMOM/AMQ0SET4') /* COBOL */ +
/* APPID('QMOM/AMQ3SET4') /* RPG - ILE */

CRTMQMPC PRCNAME('SYSTEM.SAMPLE.ECHOPROCESS') +
MQMNAME(&QMGRNAME)                        +
REPLACE(*YES)                             +
+
TEXT('Trigger process for AMQSECH4')      +
ENVDATA('JOBPTY(3)') /* Submit parameter */ +
/** Select the triggered program here **/ +
APPID('QMOM/AMQSECH4') /* C */ +
/* APPID('QMOM/AMQ0ECH4') /* COBOL */ +
/* APPID('QMOM/AMQ3ECH4') /* RPG - ILE */

/*****/
/*                               */
/* Normal return.                               */
/*                               */
/*****/
SNDPGMMSG MSG('AMQSAMP4 Completed creating sample +
objects for ' *CAT &QMGRNAME)
RETURN
ENDPGM

/*****/
/*                               */
/* END OF AMQSAMP4                               */
/*                               */
/*****/

```

## Autres méthodes d'administration de IBM MQ for IBM i

L'utilisation de commandes CL est la méthode recommandée pour l'administration de IBM MQ for IBM i. Toutefois, vous pouvez utiliser diverses autres méthodes d'administration, notamment les commandes MQSC, les commandes PCF, les commandes de contrôle et l'administration à distance.

### Pourquoi et quand exécuter cette tâche

Vous utilisez normalement les commandes CL IBM i pour administrer IBM MQ for IBM i. Pour une présentation de ces commandes, voir [«Gestion d'IBM MQ for IBM i à l'aide de commandes CL»](#), à la page 391.

Vous pouvez également utiliser des commandes MQSC et des commandes PCF, comme décrit dans les sous-rubriques, et vous pouvez utiliser des commandes de contrôle, comme décrit dans [«Administration d'IBM MQ for Multiplatforms à l'aide de commandes de contrôle»](#), à la page 11.

Vous pouvez utiliser les événements d'instrumentation IBM MQ pour surveiller le fonctionnement des gestionnaires de files d'attente. Pour plus d'informations sur les événements d'instrumentation IBM MQ et leur utilisation, voir [Événements d'instrumentation](#).

Utilisez l'une des méthodes d'administration décrites dans les sous-rubriques suivantes comme alternative à l'utilisation des commandes CL IBM i :

## Administration locale et distante sous IBM i

Vous administrez les objets IBM MQ for IBM i en local ou à distance.

### Pourquoi et quand exécuter cette tâche

*Administration locale* signifie l'exécution de tâches d'administration sur tous les gestionnaires de files d'attente que vous avez définis sur votre système local. Dans IBM MQ, vous pouvez considérer cela comme une administration locale car aucun canal IBM MQ n'est impliqué, c'est-à-dire que la communication est gérée par le système d'exploitation. Pour effectuer ce type de tâche, vous devez soit vous connecter au système distant et émettre les commandes à partir de là, soit créer un processus qui peut émettre les commandes pour vous.

IBM MQ prend en charge l'administration à partir d'un point unique via ce qu'on appelle l' *administration à distance*. L'administration à distance consiste à envoyer des messages de contrôle PCF (Programmable Command Format) à SYSTEM.ADMIN.COMMAND.QUEUE sur le gestionnaire de files d'attente cible.

Il existe plusieurs manières de générer des messages PCF. Elles sont décrites dans les étapes suivantes.

### Procédure

- Écrivez un programme à l'aide de messages PCF. Voir [«Administration à l'aide de commandes PCF sous IBM i»](#), à la page 408.
- Écrivez un programme à l'aide de MQAI, qui envoie des messages PCF. Voir [«Utilisation de MQAI pour simplifier l'utilisation des fichiers PCF»](#), à la page 40.
- Utilisez l'explorateur IBM MQ, disponible avec IBM MQ for Windows, qui vous permet d'utiliser une interface graphique et de générer les messages PCF appropriés. Voir [«Utilisation du IBM MQ Explorer avec IBM MQ for IBM i»](#), à la page 409.
- Utilisez **STRMQMQSC** pour envoyer des commandes indirectement à un gestionnaire de files d'attente éloignées. Voir [«Administration à l'aide de commandes MQSC sous IBM i»](#), à la page 407.

Par exemple, vous pouvez émettre une commande distante pour modifier une définition de file d'attente sur un gestionnaire de files d'attente éloignées.

Certaines commandes ne peuvent pas être émises de cette manière, notamment la création ou le démarrage de gestionnaires de files d'attente et de serveurs de commandes. Pour effectuer ce type de

tâche, vous devez vous connecter au système distant et émettre les commandes à partir de celui-ci ou créer un processus qui peut émettre les commandes pour vous.

IBM i

## Administration à l'aide de commandes MQSC sous IBM i

Sous IBM i, vous créez une liste de commandes dans un fichier script, puis vous exécutez le fichier à l'aide de la commande **STRMQMMQSC**. Les commandes MQSC permettent de gérer les objets de gestionnaire de files d'attente, y compris le gestionnaire de files d'attente lui-même, les files d'attente, les définitions de processus, les listes de noms, les canaux, les canaux de connexion client, les programmes d'écoute, les services, les rubriques et les objets d'informations d'authentification.

### Pourquoi et quand exécuter cette tâche

Les commandes de script IBM MQ (MQSC) sont écrites dans un format lisible par l'utilisateur, en texte EBCDIC. Vous émettez des commandes MQSC vers un gestionnaire de files d'attente à l'aide de la commande CL **STRMQMMQSC** IBM MQ. Cette méthode est une méthode de traitement par lots uniquement. Son entrée provient d'un fichier physique source du système de bibliothèque du serveur. Le nom par défaut de ce fichier physique source est QMQSC.



**Avvertissement :** N'utilisez pas la bibliothèque QTEMP comme bibliothèque source pour STRMQMMQSC, car l'utilisation de la bibliothèque QTEMP est limitée. Vous devez utiliser une autre bibliothèque comme fichier d'entrée de la commande.

Pour la portabilité entre les environnements IBM MQ, limitez la longueur de ligne dans les fichiers de commandes MQSC à 72 caractères. Utilisez le signe plus pour indiquer que la commande se poursuit sur la ligne suivante.

Les attributs d'objet spécifiés dans MQSC sont affichés dans cette rubrique en majuscules (par exemple, RQMNAME), bien qu'ils ne soient pas sensibles à la casse.

#### Remarque :

1. Le format d'un fichier MQSC ne dépend pas de son emplacement dans le système de fichiers.
2. Les noms d'attribut MQSC sont limités à huit caractères.
3. Les commandes MQSC sont disponibles sur toutes les plateformes IBM MQ.

Pour obtenir une description de chaque commande MQSC et de sa syntaxe, voir [Commandes MQSC](#).

### Procédure

1. Créez le fichier source QMQSC.

IBM MQ for IBM i ne fournit pas de fichier source appelé QMQSC. Pour traiter les commandes MQSC, vous devez créer le fichier source QMQSC dans la bibliothèque de votre choix, en exécutant la commande suivante:

```
CRTSRCPF FILE(MYLIB/QMQSC) RCDLEN(240) TEXT('IBM MQ - MQSC Source')
```

2. Gestion des membres.

La source MQSC est conservée dans les membres de ce fichier source. Pour utiliser les membres, entrez la commande suivante:

```
WRKMBRPDM MYLIB/QMQSC
```

Vous pouvez désormais ajouter de nouveaux membres et gérer des membres existants.

```

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

```

Figure 20. Extrait d'un fichier de commandes MQSC, *myprog.in*, montrant une commande MQSC (*DEFINE QLOCAL*) avec ses attributs.

### Information associée

[Administration d' IBM MQ à l'aide de commandes MQSC](#)

## IBM i Administration à l'aide de commandes PCF sous IBM i

L'objectif des commandes PCF (Programmable Command Format) IBM MQ est de permettre la programmation des tâches d'administration dans un programme d'administration. Vous pouvez ainsi créer des files d'attente et des définitions de processus, et modifier des gestionnaires de files d'attente à partir d'un programme.

Les commandes PCF couvrent la même gamme de fonctions que les commandes MQSC. Cependant, contrairement aux commandes MQSC, les commandes PCF et leurs réponses ne sont pas dans un format texte que vous pouvez lire.

Vous pouvez écrire un programme pour émettre des commandes PCF vers n'importe quel gestionnaire de files d'attente du réseau à partir d'un noeud unique. De cette manière, vous pouvez à la fois centraliser et automatiser les tâches d'administration.

Chaque commande PCF est une structure de données imbriquée dans la partie données d'application d'un message IBM MQ . Chaque commande est envoyée au gestionnaire de files d'attente cible à l'aide de la fonction MQI MQPUT de la même manière que tout autre message. Le serveur de commandes du gestionnaire de files d'attente qui reçoit le message l'interprète comme un message de commande et exécute la commande. Pour obtenir les réponses, l'application émet un appel MQGET et les données de réponse sont renvoyées dans une autre structure de données. L'application peut ensuite traiter la réponse et agir en conséquence.

En bref, voici quelques-uns des éléments que le programmeur d'application doit spécifier pour créer un message de commande PCF:

### Descripteur de message

Il s'agit d'un descripteur de message IBM MQ standard, dans lequel:

- Le type de message (*MsgType*) est MQMT\_REQUEST.
- Le format de message (*Format*) est MQFMT\_ADMIN.

### Données d'application

Contient le message PCF incluant l'en-tête PCF, dans lequel:

- Le type de message PCF (*Type*) indique MQCFT\_COMMAND.
- L'identificateur de la commande indique la commande, par exemple *Change Queue* (MQCMD\_CHANGE\_Q).

Les fichiers PCF d'arrêt programme sont des commandes PCF qui contiennent des commandes MQSC dans le texte du message. Vous pouvez utiliser des fichiers PCF pour envoyer des commandes à



un gestionnaire de files d'attente éloignées. Pour plus d'informations, voir [«Utilisation de MQAI pour simplifier l'utilisation des fichiers PCF»](#), à la page 40.

Pour obtenir une description complète des structures de données PCF et savoir comment les implémenter, voir [Structures des commandes et des réponses](#).

## **Utilisation du IBM MQ Explorer avec IBM MQ for IBM i**

Utilisez ces informations pour administrer IBM MQ for IBM i à l'aide de la IBM MQ Explorer.

IBM MQ for Windows (plateformex86 ) et IBM MQ for Linux (plateformesx86 et x86-64 ) fournissent une interface d'administration appelée IBM MQ Explorer pour effectuer des tâches d'administration au lieu d'utiliser des commandes CL, de contrôle ou MQSC.

Le IBM MQ Explorer vous permet d'effectuer l'administration locale ou à distance de votre réseau à partir d'un ordinateur exécutant Windows (plateformex86 ) ou Linux (plateformesx86 et x86-64 ), en pointant le IBM MQ Explorer sur les gestionnaires de files d'attente et les clusters qui vous intéressent.

Avec IBM MQ Explorer, vous pouvez:

- Démarrez et arrêtez un gestionnaire de files d'attente (sur votre machine locale uniquement).
- Définissez, affichez et modifiez les définitions des objets IBM MQ tels que les files d'attente, les rubriques et les canaux.
- Parcourez les messages d'une file d'attente.
- Démarrer et arrêter un canal.
- Afficher les informations de statut d'un canal.
- Afficher les gestionnaires de files d'attente dans un cluster.
- Vérifiez les applications, les utilisateurs ou les canaux pour lesquels une file d'attente particulière est ouverte.
- Créez un cluster de gestionnaires de files d'attente à l'aide de l'assistant **Créer un cluster** .
- Ajoutez un gestionnaire de files d'attente à un cluster à l'aide de l'assistant **Ajout d'un gestionnaire de files d'attente à un cluster** .
- Gérez l'objet d'informations d'authentification, utilisé avec la sécurité du canal TLS (Transport Layer Security).

En utilisant les conseils en ligne, vous pouvez:

- Définissez et contrôlez diverses ressources, notamment les gestionnaires de files d'attente, les files d'attente, les canaux, les définitions de processus, les canaux de connexion client, les programmes d'écoute, les rubriques, les services, les listes de noms et les clusters.
- Démarrer ou arrêter un gestionnaire de files d'attente et ses processus associés.
- Affichez les gestionnaires de files d'attente et les objets associés sur votre poste de travail ou à partir d'autres postes de travail.
- Vérifiez le statut des gestionnaires de files d'attente, des clusters et des canaux.

Vérifiez que vous avez satisfait les exigences suivantes avant de tenter d'utiliser IBM MQ Explorer pour gérer IBM MQ sur une machine serveur. Vérifiez que:

1. Un serveur de commandes est en cours d'exécution pour tous les gestionnaires de files d'attente administrés, démarrés sur le serveur par la commande CL **STRMQMSVR**.
2. Un programme d'écoute TCP/IP approprié existe pour chaque gestionnaire de files d'attente éloignées. Il s'agit du programme d'écoute IBM MQ démarré par la commande **STRMQMLSR** .
3. Le canal de connexion serveur, appelé SYSTEM . ADMIN . SVRCONN, existe sur chaque gestionnaire de files d'attente éloignées. Vous devez créer ce canal vous-même. Il est obligatoire pour chaque gestionnaire de files d'attente éloignées administré. Sans elle, l'administration à distance n'est pas possible.
4. Vérifiez que la file d'attente SYSTEM . MQEXPLORER . REPLY . MODEL existe.

## Gestion du serveur de commandes pour l'administration à distance sous IBM i

Utilisez ces informations pour en savoir plus sur l'administration à distance du serveur de commandes IBM MQ for IBM i .

Chaque gestionnaire de files d'attente peut être associé à un serveur de commandes. Un serveur de commandes traite toutes les commandes entrantes provenant des gestionnaires de files d'attente éloignées ou les commandes PCF provenant des applications. Il présente les commandes au gestionnaire de files d'attente pour le traitement et renvoie un code achèvement ou un message d'opérateur en fonction de l'origine de la commande.

Un serveur de commandes est obligatoire pour toute administration impliquant des fichiers PCF, MQAI, ainsi que pour l'administration à distance.

**Remarque :** Pour l'administration à distance, vous devez vous assurer que le gestionnaire de files d'attente cible est en cours d'exécution. Sinon, les messages contenant des commandes ne peuvent pas quitter le gestionnaire de files d'attente à partir duquel elles sont émises. A la place, ces messages sont mis en file d'attente dans la file d'attente de transmission locale qui sert le gestionnaire de files d'attente éloignées. Dans la mesure du possible, évitez cette situation.

Il existe des commandes de contrôle distinctes pour le démarrage et l'arrêt du serveur de commandes. Vous pouvez effectuer les opérations décrites dans les sections suivantes à l'aide de l'explorateur IBM MQ .

### Démarrage et arrêt du serveur de commandes

Pour démarrer le serveur de commandes, utilisez la commande CL suivante:

```
STRMQCSVR MQMNAME('saturn.queue.manager')
```

où `saturn.queue.manager` est le gestionnaire de files d'attente pour lequel le serveur de commandes est démarré.

Pour arrêter le serveur de commandes, utilisez l'une des commandes CL suivantes:

1. 

```
ENDMQCSVR MQMNAME('saturn.queue.manager') OPTION(*CNTRLD)
```

pour effectuer un arrêt contrôlé, où `saturn.queue.manager` correspond au gestionnaire de files d'attente pour lequel le serveur de commandes est arrêté. Il s'agit de l'option par défaut, qui signifie que `OPTION(*CNTRLD)` peut être omis.

2. 

```
ENDMQCSVR MQMNAME('saturn.queue.manager') OPTION(*IMMED)
```

pour effectuer un arrêt immédiat, où `saturn.queue.manager` correspond au gestionnaire de files d'attente pour lequel le serveur de commandes est arrêté.

### Affichage de l'état du serveur de commandes

Pour l'administration à distance, vérifiez que le serveur de commandes sur le gestionnaire de files d'attente cible est en cours d'exécution. S'il n'est pas en cours d'exécution, les commandes distantes ne peuvent pas être traitées. Tous les messages contenant des commandes sont mis en file d'attente dans la file d'attente de commandes du gestionnaire de files d'attente cible `SYSTEM.ADMIN.COMMAND.QUEUE`.

Pour afficher l'état du serveur de commandes d'un gestionnaire de files d'attente, appelé ici `saturn.queue.manager`, la commande CL est la suivante:

```
DSPMQCSVR MQMNAME('saturn.queue.manager')
```

Exécutez cette commande sur la machine cible. Si le serveur de commandes est en cours d'exécution, le panneau présenté dans la [Figure 21](#), à la page 411 apparaît:

```
Display MQM Command Server (DSPMQCSVR)

Queue manager name . . . . . > saturn.queue.manager
MQM Command Server Status. . . . > RUNNING

F3=Exit  F4=Prompt  F5=Refresh  F12=Cancel  F13=How to use this display
F24=More keys
```

Figure 21. Afficher le panneau MQM Command Server

## IBM i Exécution des commandes de la console Web

Vous devez configurer votre environnement comme décrit dans le texte suivant pour que les commandes Qshell associées à la console Web s'exécutent correctement sous IBM MQ for IBM i.

### Pourquoi et quand exécuter cette tâche

Lorsque le Qshell démarre, il initialise les tables internes pour le traitement des commandes en fonction du CCSID du travail. Pour que les commandes Qshell associées à la console Web s'exécutent correctement sous IBM i, vous devez configurer votre environnement.

Un environnement local est défini en définissant la variable d'environnement LANG sur le nom de chemin d'accès à un objet d'environnement local. Par exemple, pour définir l'environnement local pour l'anglais américain, la variable d'environnement LANG est définie comme suit:

```
LANG=/QSYS.LIB/EN_US.LOCALE
```

Dans Qshell, vous pouvez vérifier le paramètre en exécutant un ensemble de commandes pour répertorier toutes les variables d'environnement. En général, il s'agit de LANG qui peut avoir un impact sur l'environnement local de l'environnement d'exécution. Il peut également avoir LC\_ALL.

Pour exécuter correctement les commandes Qshell, le paramètre d'environnement local doit être cohérent avec votre paramètre de travail.

### Procédure

Utilisez la commande CL DSPJOB JOB (JobNumber/USERProfile/JobName)

- a) Sélectionnez l'option 2 pour afficher les attributs de définition de travail.
- b) Les attributs suivants doivent être cohérents avec le paramètre d'environnement LANG ou LC\_ALL
  - Identificateur de langue
  - Identificateur de pays ou de région
  - Identificateur de jeu de caractères codés

Par exemple, si

```
LANG=/QSYS.LIB/FR_FR.LOCALE
```

vos attributs de travail doivent être:

- ID langue..... FRA

- ID pays ou région..... FR
- ID de jeu de caractères codés... 297

## Que faire ensuite

Pour plus d'informations sur le support de langue nationale, voir la rubrique IBM Documentation [Remarques sur le support de langue nationale \(NLS\)](#).

## IBM i Gestion des travaux pour IBM i

Ces informations décrivent la manière dont IBM MQ traite les demandes de travail et détaillent les options disponibles pour hiérarchiser et contrôler les travaux associés à IBM MQ.

### Avertissement

Ne modifiez pas les objets de gestion des travaux IBM MQ à moins que vous ne compreniez parfaitement les concepts de la gestion des travaux IBM i et IBM MQ .

Des informations supplémentaires sur les sous-systèmes et les descriptions de travail sont disponibles sous [Gestion des travaux](#) dans la documentation du produit IBM i . Prêtez une attention particulière aux sections [Démarrage des travaux](#) et [Travaux par lots](#).

IBM MQ for IBM i intègre l'environnement IBM i UNIX et les unités d'exécution IBM i . N'apportez **pas** de modifications aux objets du Integrated File System (IFS).

Lors des opérations normales, un gestionnaire de files d'attente IBM MQ démarre un certain nombre de travaux par lots pour effectuer différentes tâches. Par défaut, ces travaux par lots s'exécutent dans le sous-système QMQM créé lors de l'installation de IBM MQ .

La gestion des travaux fait référence au processus de personnalisation des tâches IBM MQ pour obtenir les performances optimales de votre système ou pour simplifier l'administration.

Par exemple, vous pouvez effectuer les tâches suivantes :

- Modifiez la priorité d'exécution des travaux pour rendre un gestionnaire de files d'attente plus réactif qu'un autre.
- Redirige la sortie d'un certain nombre de travaux vers une file d'attente en sortie spécifique.
- Faites en sorte que tous les travaux d'un certain type soient exécutés dans un sous-système spécifique.
- Isolez les erreurs dans un sous-système.

La gestion des travaux s'effectue en créant ou en modifiant les descriptions de travail associées aux travaux IBM MQ . Vous pouvez configurer la gestion des travaux pour:

- Une installation IBM MQ complète.
- Gestionnaires de files d'attente individuels.
- Travaux individuels pour des gestionnaires de files d'attente individuels.

## IBM i Tâches IBM MQ pour IBM i

Il s'agit d'un tableau des travaux IBM MQ for IBM i et d'une brève description de chacun d'eux.

Lorsqu'un gestionnaire de files d'attente est en cours d'exécution, vous voyez certains ou tous les travaux par lots suivants s'exécutant sous le profil utilisateur QMQM dans le sous-système IBM MQ . Les travaux sont décrits brièvement dans [Tableau 21](#), à la page 413.

Vous pouvez afficher tous les travaux connectés à un gestionnaire de files d'attente à l'aide de l'option 22 du panneau **Work with Queue Manager** (WRKMQM). Vous pouvez afficher les programmes d'écoute à l'aide de la commande WRKMQLSR.

Tableau 21. Tâches IBM MQ .

Nom du travail	Fonction
AMQZMUC0	Gestionnaire d'utilitaires. Ce travail exécute des utilitaires de gestionnaire de files d'attente critiques, par exemple le gestionnaire de chaînes de journaux.
AMQZXMA0	Contrôleur d'exécution correspondant au premier travail démarré par le gestionnaire de files d'attente. Il traite les demandes MQCONN et démarre les processus d'agent pour traiter les appels API IBM MQ .
AMQZFUMA	Gestionnaire des droits d'accès aux objets (OAM).
AMQZLAA0	Les agents de gestionnaire de files d'attente qui effectuent la plupart du travail pour les applications qui se connectent au gestionnaire de files d'attente à l'aide de MQCNO_STANDARD_BINDING.
AMQZLSA0	Agent de gestionnaire de files d'attente.
AMQZMUFO	Gestionnaire d'utilitaire
AMQZMGRO	Contrôleur de processus. Ce travail est utilisé pour démarrer et gérer les programmes d'écoute et les services.
AMQZMURO	Gestionnaire d'utilitaires. Ce travail exécute des utilitaires de gestionnaire de files d'attente critiques, par exemple le gestionnaire de chaînes de journaux.
AMQFQPUB	Démon de publication / abonnement en file d'attente.
AMQFCXBA	Travail d'agent de courtier.
RUNMQBRK	Travail de contrôle du courtier.
AMQRMPPA	Travail de regroupement de processus de canal.
amqcrsta	Canal répondeur appelé par TCP/IP.
AMQCRS6B	Canal récepteur LU62 et connexion client (voir remarque).
AMQRRMFA	Gestionnaire de référentiels pour les clusters.
AMQCLMAA	Programme d'écoute TCP/IP sans unités d'exécution.
AMQPCSEA	Processeur de commandes PCF qui gère les demandes PCF et d'administration à distance.
runmqtrm	Moniteur de déclenchement.
runmqdlq	Gestionnaire de file d'attente de rebut.
runmqchi	Initiateur de canal.
runmqchl	Travail de canal émetteur démarré pour chaque canal émetteur.
runmqlsr	Programme d'écoute TCP/IP à unités d'exécution.
AMQRCMLA	Canal MQSC et processeur de commandes PCF.

**Remarque :** Le travail récepteur LU62 s'exécute dans le sous-système de communications et extrait ses propriétés d'exécution des entrées de routage et de communication utilisées pour démarrer le travail. Pour plus d'informations, voir [Initiated end \(Receiver\)](#) .

## Objets de gestion des travaux sous IBM i

Lorsque IBM MQ est installé, divers objets sont fournis dans la bibliothèque QMQM pour faciliter la gestion des travaux. Ces objets sont ceux dont les travaux IBM MQ ont besoin pour s'exécuter dans leur propre sous-système.

Des exemples de descriptions de travail sont fournis pour deux des travaux par lots IBM MQ . Si aucune description de travail spécifique n'est fournie pour un travail IBM MQ , elle s'exécute avec la description de travail par défaut QMQMJOB.

Les objets de gestion des travaux qui sont fournis lorsque vous installez IBM MQ sont répertoriés dans [Tableau 22, à la page 414](#) et les objets créés pour un gestionnaire de files d'attente sont répertoriés dans [Tableau 23, à la page 414](#)

**Remarque :** Les objets de gestion des travaux se trouvent dans la bibliothèque QMQM et les objets de gestionnaire de files d'attente se trouvent dans la bibliothèque du gestionnaire de files d'attente.

Nom	Tapez	Description
AMQZLAA0	*JOB	Description de travail utilisée par les processus de l'agent IBM MQ
AMQZLSA0	*JOB	Agent de gestionnaire de files d'attente de liaisons isolées
AMQZXMA0	*JOB	Description de travail utilisée par les contrôleurs d'exécution IBM MQ
QMQM	*SBSD	Sous-système dans lequel tous les travaux IBM MQ s'exécutent
QMQM	*JOBQ	File d'attente de travaux connectée au sous-système fourni
QMQMJOB	*JOB	Description de travail IBM MQ par défaut, utilisée s'il n'existe pas de description de travail spécifique pour un travail
QMQMMSG	*MSGQ	File d'attente de messages par défaut pour les travaux IBM MQ .
QMQMRUN20	*CLS	Description de classe pour les travaux IBM MQ à priorité élevée
QMQMRUN35	*CLS	Description de classe pour les travaux IBM MQ de priorité moyenne
QMQMRUN50	*CLS	Description de classe pour les travaux IBM MQ de priorité basse

Nom	Tapez	Description
AMQA000000	*JRNRCV	Récepteur de journal local
AMQAJRN	*JRN	Journal local
AMQJRNINF	*USRSPC	Espace utilisateur mis à jour avec les derniers récepteurs de journal requis pour le démarrage et la reprise sur support d'un gestionnaire de files d'attente. Cet espace utilisateur peut être interrogé par une application pour déterminer quels récepteurs de journal doivent être archivés et lesquels peuvent être supprimés en toute sécurité.
AMQAJRNMSG	*MSGQ	File d'attente de messages du journal local
AMQCRC6B	*PGM	Programme de démarrage de la connexion LU6.2
AMQRFOLD	*FILE	Fichier de définition de canal de gestionnaire de files d'attente migré
QMQMMSG	*MSGQ	File de messages du gestionnaire de files d'attente

**IBM i**

Ces informations décrivent la manière dont IBM MQ utilise les objets de gestion des travaux et fournissent des exemples de configuration.



**Avertissement :** Ne modifiez pas les paramètres des postes de file d'attente de travaux dans le sous-système QMQM pour limiter le nombre de travaux autorisés dans le sous-système par priorité. Si vous tentez d'effectuer cette opération, vous pouvez arrêter l'exécution des travaux IBM MQ essentiels une fois qu'ils ont été soumis et provoquer l'échec du démarrage du gestionnaire de files d'attente.

Pour comprendre comment configurer la gestion des travaux, vous devez d'abord comprendre comment IBM MQ utilise les descriptions de travail.

La description de travail utilisée pour démarrer le travail contrôle de nombreux attributs du travail.

Exemple :

- File d'attente de travaux dans laquelle le travail est mis en file d'attente et sur quel sous-système le travail est exécuté.
- Données de routage utilisées pour démarrer le travail et la classe utilisés par le travail pour ses paramètres d'exécution.
- File d'attente en sortie utilisée par le travail pour les fichiers d'impression.

Le processus de démarrage d'un travail IBM MQ peut être considéré en trois étapes:

1. IBM MQ sélectionne une description de travail.

IBM MQ utilise la technique suivante pour déterminer la description de travail à utiliser pour un travail par lots:

- a. Recherchez dans la bibliothèque du gestionnaire de files d'attente une description de travail portant le même nom que le travail. Voir [Présentation des noms de bibliothèque de gestionnaire de files d'attente IBM MQ for IBM i](#) pour plus de détails sur la bibliothèque de gestionnaire de files d'attente.
- b. Recherchez la description de travail par défaut QMQMJOB dans la bibliothèque du gestionnaire de files d'attente.
- c. Recherchez dans la bibliothèque QMQM une description de travail portant le même nom que le travail.
- d. Utilisez la description de travail par défaut, QMQMJOB, dans la bibliothèque QMQM.

2. Le travail est soumis à la file d'attente de travaux.

Les descriptions de travail fournies avec IBM MQ ont été configurées, par défaut, pour placer des travaux dans la file d'attente de travaux QMQM dans la bibliothèque QMQM. La file d'attente de travaux QMQM étant connectée au sous-système QMQM fourni, les travaux démarrent par défaut dans le sous-système QMQM.

3. Le travail entre dans le sous-système et passe par les phases d'activation.

Lorsque le travail entre dans le sous-système, les données de routage indiquées dans la description de travail sont utilisées pour rechercher des postes de routage pour le travail.

Les données de routage doivent correspondre à l'une des entrées de routage définies dans le sous-système QMQM, ce qui définit les classes fournies (QMQRUN20, QMQRUN35 ou QMQRUN50) qui sont utilisées par le travail.

**Remarque :** Si les travaux IBM MQ ne semblent pas démarrer, assurez-vous que le sous-système est en cours d'exécution et que la file d'attente de travaux n'est pas suspendue,

Si vous avez modifié les objets de gestion des travaux IBM MQ, assurez-vous que tout est correctement associé. Par exemple, si vous spécifiez une file d'attente de travaux autre que QMQM/QMQM dans la

description de travail, assurez-vous qu'une opération ADDJOBQE est effectuée pour le sous-système, c'est-à-dire QMQM.

Vous pouvez créer une description de travail pour chaque travail documenté dans [Tableau 21](#), à la page 413 en utilisant la feuille de travail suivante comme exemple:

```
What is the queue manager library name? _____
Does job description AMQZXMA0 exist in the queue manager library? Yes No
Does job description QMQMJOB0 exist in the queue manager library? Yes No
Does job description AMQZXMA0 exist in the QMQM library? Yes No
Does job description QMQMJOB0 exist in the QMQM library? Yes No
```

Si vous répondez Non à toutes ces questions, créez une description de travail globale QMQMJOB0 dans la bibliothèque QMQM.

## File d'attente de messages IBM MQ

Une file d'attente de messages IBM MQ , QMQMMSG, est créée dans chaque bibliothèque de gestionnaire de files d'attente. Les messages du système d'exploitation sont envoyés à cette file d'attente lorsque les travaux du gestionnaire de files d'attente s'arrêtent et IBM MQ envoie des messages à la file d'attente. Par exemple, pour signaler les récepteurs de journal nécessaires au démarrage. Conservez le nombre de messages dans cette file d'attente de messages à une taille gérable afin de faciliter la surveillance.

### IBM i Exemples de système par défaut pour IBM i

Ces exemples montrent comment fonctionne une installation IBM MQ non modifiée lorsque certains des travaux standard sont soumis au démarrage du gestionnaire de files d'attente.

Tout d'abord, le travail du contrôleur d'exécution AMQZXMA0 démarre.

1. Exécutez la commande **STRMQM** pour le gestionnaire de files d'attente TESTQM.
2. IBM MQ recherche dans la bibliothèque du gestionnaire de files d'attente QMTESTQM, d'abord la description de travail AMQZXMA0, puis la description de travail QMQMJOB0.

Comme aucune de ces descriptions de travail n'existe, IBM MQ recherche la description de travail AMQZXMA0 dans la bibliothèque de produits QMQM. Cette description de travail existe, elle est donc utilisée pour soumettre le travail.

3. La description de travail utilise la file d'attente de travaux par défaut IBM MQ , de sorte que le travail est soumis à la file d'attente de travaux QMQM/QMQM.
4. Les données de routage de la description de travail AMQZXMA0 étant QMQMRUN20, le système recherche les entrées de routage de sous-système correspondant à ces données.

Par défaut, le poste de routage avec le numéro de séquence 9900 possède des données de comparaison qui correspondent à QMQMRUN20, de sorte que le travail est démarré avec la classe définie sur ce poste de routage, également appelé QMQMRUN20.

5. La priorité d'exécution de la classe QMQM/QMQMRUN20 étant définie sur 20, le travail AMQZXMA0 s'exécute dans le sous-système QMQM avec la même priorité que la plupart des travaux interactifs sur le système.

### IBM i Configuration d'exemples de gestion des travaux sous IBM i

Utilisez ces informations pour savoir comment modifier et créer des descriptions de travail IBM MQ afin de modifier les attributs d'exécution des travaux IBM MQ .

La clé de la flexibilité de la gestion des travaux IBM MQ réside dans la manière à deux niveaux dont IBM MQ recherche les descriptions de travail:

- Si vous créez ou modifiez des descriptions de travail dans une bibliothèque de gestionnaire de files d'attente, ces modifications remplacent les descriptions de travail globales dans QMQM, mais les modifications sont locales et affectent uniquement ce gestionnaire de files d'attente particulier.



- Si vous créez ou modifiez des descriptions de travail globales dans la bibliothèque QMQM, ces descriptions de travail affectent tous les gestionnaires de files d'attente du système, sauf si elles sont remplacées localement pour des gestionnaires de files d'attente individuels.
1. L'exemple suivant augmente la priorité des travaux de contrôle de canal pour un gestionnaire de files d'attente individuel.

Pour que les travaux du gestionnaire de référentiels et de l'initiateur de canal, AMQRRMFA et RUNMQCHI, soient exécutés aussi rapidement que possible pour le gestionnaire de files d'attente TESTQM, procédez comme suit:

- a. Créez des doublons locaux de la description de travail QMQM/QMQMJOBDD avec les noms des processus IBM MQ que vous souhaitez contrôler dans la bibliothèque du gestionnaire de files d'attente. Exemple :

```
CRTDUPOBJ OBJ(QMQMJOBDD) FROMLIB(QMQM) OBJTYPE(*JOBDD) TOLIB(QMTESTQM)
NEWOBJ(RUNMQCHI)
CRTDUPOBJ OBJ(QMQMJOBDD) FROMLIB(QMQM) OBJTYPE(*JOBDD) TOLIB(QMTESTQM)
NEWOBJ(AMQRRMFA)
```

- b. Modifiez le paramètre de données de routage dans la description de travail pour vous assurer que les travaux utilisent la classe QMQMRUN20 .

```
CHGJOBDD JOBDD(QMTESTQM/RUNMQCHI) RTGDTA('QMQMRUN20')
CHGJOBDD JOBDD(QMTESTQM/AMQRRMFA) RTGDTA('QMQMRUN20')
```

Les travaux AMQRRMFA et RUNMQCHI pour le gestionnaire de files d'attente TESTQM sont maintenant:

- Utiliser les nouvelles descriptions de travail locales dans la bibliothèque du gestionnaire de files d'attente
  - S'exécute avec la priorité 20, car la classe QMQMRUN20 est utilisée lorsque les travaux entrent dans le sous-système.
2. L'exemple suivant définit une nouvelle classe de priorité d'exécution pour le sous-système QMQM.
    - a. Créez une classe en double dans la bibliothèque QMQM pour permettre à d'autres gestionnaires de files d'attente d'accéder à la classe, en exécutant la commande suivante:

```
CRTDUPOBJ OBJ(QMQMRUN20) FROMLIB(QMQM) OBJTYPE(*CLS) TOLIB(QMQM)
NEWOBJ(QMQMRUN10)
```

- b. Modifiez la classe pour qu'elle ait la nouvelle priorité d'exécution en exécutant la commande suivante:

```
CHGCLS CLS(QMQM/QMQMRUN10) RUNPTY(10)
```

- c. Ajoutez la nouvelle définition de classe au sous-système en exécutant la commande suivante:

```
ADDRTGE SBSDD(QMQM/QMQM) SEQNBR(8999) CMPVAL('QMQMRUN10') PGM(QSYS/QCMD)
CLS(QMQM/QMQMRUN10)
```

**Remarque :** Vous pouvez indiquer n'importe quelle valeur numérique pour le numéro de séquence de routage, mais les valeurs doivent être séquentielles. Ce numéro de séquence indique au sous-système l'ordre dans lequel les postes de routage doivent être recherchés pour une correspondance de données de routage.

- d. Modifiez la description de travail locale ou globale pour utiliser la nouvelle classe de priorité en exécutant la commande suivante:

```
CHGJOBDD JOBDD(QMQMlibname/QMQMJOBDD) RTGDTA('QMQMRUN10')
```

Tous les travaux de gestionnaire de files d'attente associés à QMLibraryname utilisent désormais une priorité d'exécution de 10.

### 3. L'exemple suivant exécute un gestionnaire de files d'attente dans son propre sous-système

Pour que tous les travaux du gestionnaire de files d'attente TESTQM soient exécutés dans le sous-système QBATCH, procédez comme suit:

- a. Créez un doublon local de la description de travail QMQM/QMQMJOBDB dans la bibliothèque du gestionnaire de files d'attente à l'aide de la commande

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMTESTQM)
```

- b. Modifiez le paramètre de file d'attente de travaux dans la description de travail pour vous assurer que les travaux utilisent la file d'attente de travaux QBATCH.

```
CHGJOBDB JOBDB(QMTESTQM/QMQMJOBDB) JOBQ(*LIBL/QBATCH)
```

**Remarque :** La file d'attente de travaux est associée à la description de sous-système. Si vous constatez que les travaux restent dans la file d'attente de travaux, vérifiez que la définition de la file d'attente de travaux est définie sur le SBSDB. Utilisez la commande DSPSBSDB pour le sous-système et choisissez l'option 6, Postes de file d'attente de travaux.

Tous les travaux du gestionnaire de files d'attente TESTQM sont maintenant:

- Utiliser la nouvelle description de travail par défaut locale dans la bibliothèque du gestionnaire de files d'attente
- Sont soumis à la file d'attente de travaux QBATCH.

Pour vous assurer que les travaux sont routés et hiérarchisés correctement:

- Créez des entrées de routage pour les travaux IBM MQ dans le sous-système QBATCH ou
- Utilisez une entrée de routage fourre-tout qui appelle QCMD, quelles que soient les données de routage utilisées.

Cette option ne fonctionne que si l'option Nombre maximal de travaux actifs pour la file d'attente de travaux QBATCH est définie sur \*NOMAX. La valeur par défaut du système est 1.

### 4. L'exemple suivant crée un autre sous-système IBM MQ

- a. Créez un sous-système en double dans la bibliothèque QMQM en exécutant la commande suivante:

```
CRTDUPOBJ OBJ(QMQM) FROMLIB(QMQM) OBJTYPE(*SBSDB) TOLIB(QMQM) NEWOBJ(QMQM2)
```

- b. Supprimez la file d'attente de travaux QMQM en exécutant la commande suivante:

```
RMVJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM)
```

- c. Créez une file d'attente de travaux pour le sous-système en exécutant la commande suivante:

```
CRTJOBQ JOBQ(QMQM/QMQM2) TEXT('Job queue for IBM MQ Queue Manager')
```

- d. Ajoutez un poste file d'attente de travaux au sous-système en exécutant la commande suivante:

```
ADDJOBQE SBSDB(QMQM/QMQM2) JOBQ(QMQM/QMQM2) MAXACT(*NOMAX)
```

- e. Créez un QMQMJOBDB en double dans la bibliothèque du gestionnaire de files d'attente en exécutant la commande suivante:

```
CRTDUPOBJ OBJ(QMQMJOBDB) FROMLIB(QMQM) OBJTYPE(*JOBDB) TOLIB(QMLibraryname)
```

- f. Modifiez la description de travail pour utiliser la nouvelle file d'attente de travaux à l'aide de la commande suivante:

```
CHGJOB JOB(QMlibaryname/QMQMJOB) JOBQ(QMQM/QMQM2)
```

- g. Démarrez le sous-système en exécutant la commande suivante:

```
STRSBS SBS(QMQM/QMQM2)
```

**Remarque :**

- Vous pouvez spécifier le sous-système dans n'importe quelle bibliothèque. Si, pour une raison quelconque, le produit est réinstallé ou que la bibliothèque QMQM est remplacée, les modifications apportées sont supprimées.
- Tous les travaux de gestionnaire de files d'attente associés à QMlibaryname sont désormais exécutés sous le sous-système QMQM2.

**IBM i**

## Disponibilité, sauvegarde, récupération et redémarrage sur IBM i

Utilisez ces informations pour comprendre comment IBM MQ for IBM i utilise le support de journalisation IBM i pour aider sa stratégie de sauvegarde et de restauration.

Avant de lire cette section, vous devez connaître les méthodes de sauvegarde et de reprise IBM i standard, ainsi que l'utilisation des journaux et des récepteurs de journal associés sous IBM i. Pour plus d'informations sur ces rubriques, voir [Sauvegarde et reprise](#).

Pour comprendre la stratégie de sauvegarde et de reprise, vous devez d'abord comprendre comment IBM MQ for IBM i organise ses données dans le système de fichiers IBM i et le système de fichiers intégré (IFS).

IBM MQ for IBM i stocke ses données dans une bibliothèque individuelle pour chaque instance de gestionnaire de files d'attente et dans les fichiers STREAM du système de fichiers IFS.

Les bibliothèques spécifiques du gestionnaire de files d'attente contiennent les journaux, les récepteurs de journal et les objets requis pour contrôler la gestion des travaux du gestionnaire de files d'attente. Les répertoires et fichiers IFS contiennent les fichiers de configuration IBM MQ, les descriptions des objets IBM MQ et les données qu'ils contiennent.

Chaque modification apportée à ces objets, qui peut être récupérée en cas de défaillance du système, est enregistrée dans un journal *avant* qu'elle ne soit appliquée à l'objet approprié. Cela a pour effet que ces modifications peuvent être récupérées en réalisant les informations enregistrées dans le journal.

Vous pouvez configurer IBM MQ for IBM i pour utiliser plusieurs instances de gestionnaire de files d'attente sur différents serveurs afin d'augmenter la disponibilité du gestionnaire de files d'attente et d'accélérer la reprise en cas de défaillance d'un serveur ou d'un gestionnaire de files d'attente.

**IBM i**

## Journaux de gestionnaire de files d'attente sous IBM i

Utilisez ces informations pour comprendre comment IBM MQ for IBM i utilise les journaux dans son fonctionnement pour contrôler les mises à jour des objets locaux.

Chaque bibliothèque de gestionnaire de files d'attente contient un journal pour ce gestionnaire de files d'attente, et le journal est nommé QM *GRLIB*/AMQ *A* JRN, où QM *GRLIB* est le nom de la bibliothèque de gestionnaire de files d'attente, et *A* est une lettre, *A* dans le cas d'un gestionnaire de files d'attente à instance unique, qui est unique à l'instance de gestionnaire de files d'attente.

QM *GRLIB* prend le nom QM, suivi du nom du gestionnaire de files d'attente sous une forme unique. Par exemple, un gestionnaire de files d'attente nommé TEST possède une bibliothèque de gestionnaire de files d'attente nommée QMTEST. La bibliothèque du gestionnaire de files d'attente peut être spécifiée lors de la création d'un gestionnaire de files d'attente à l'aide de la commande **CRTMQM**.

Les journaux sont associés à des récepteurs de journal qui contiennent les informations en cours de journalisation. Les récepteurs sont des objets auxquels les informations peuvent uniquement être ajoutées et qui finiront par se remplir.

Les récepteurs de journal utilisent un espace disque précieux avec des informations obsolètes. Toutefois, vous pouvez placer les informations dans la mémoire permanente pour réduire ce problème. Un récepteur de journal est attaché au journal à un moment donné. Si le récepteur de journal atteint sa taille de seuil prédéterminée, il est détaché et remplacé par un nouveau récepteur de journal. Vous pouvez spécifier le seuil des récepteurs de journal lorsque vous créez un gestionnaire de files d'attente à l'aide de **CRTMQM** et du paramètre **THRESHOLD**.

Les récepteurs de journal associés au journal IBM MQ for IBM i local existent dans chaque bibliothèque de gestionnaire de files d'attente et adoptent une convention de dénomination comme suit:

```
AMQ Arnnnnn
```

Où

**A**

est une lettre A-Z. Il s'agit de A pour les gestionnaires de files d'attente à instance unique. Il varie selon les instances d'un gestionnaire de files d'attente multi-instance.

**nnnnn**

est la valeur décimale 00000 to 99999 qui est incrémentée de 1 pour le journal suivant dans la séquence.

**r**

est la valeur décimale 0 to 9, qui est incrémentée de 1 chaque fois qu'un récepteur est restauré.

La séquence des journaux est basée sur la date. Toutefois, la désignation du journal suivant est basée sur les règles suivantes:

1. AMQA $r$ nnnnn passe à AMQA $r$ (nnnnn+1) et nnnnn effectue un retour à la ligne lorsqu'il atteint 99999. Par exemple, AMQA099999 passe à AMQA000000 et AMQA999999 passe à AMQA900000.
2. Si un journal portant un nom généré par la règle 1 existe déjà, le message CPI70E3 est envoyé à la file d'attente de messages QSYSOPR et la commutation automatique de récepteurs s'arrête.

Le récepteur actuellement attaché continue d'être utilisé jusqu'à ce que vous examiniez l'incident et que vous attaquiez manuellement un nouveau récepteur.

3. Si aucun nouveau nom n'est disponible dans la séquence (c'est-à-dire que tous les noms de journal possibles se trouvent sur le système), vous devez effectuer les deux opérations suivantes:
  - a. Supprimez les journaux dont vous n'avez plus besoin (voir [«Gestion des journaux sous IBM i»](#), à la page 425).
  - b. Enregistrez les modifications de journal dans le récepteur de journal le plus récent à l'aide de (**RCDMQMIMG**) puis répétez l'étape précédente. Cela permet de réutiliser les anciens noms de récepteurs de journal.

Le journal AMQAJRN utilise l'option MNGRCV(\*SYSTEM) pour permettre au système d'exploitation de modifier automatiquement les récepteurs de journal lorsque le seuil est atteint. Pour plus d'informations sur la manière dont le système gère les récepteurs, voir *IBM i Backup and Recovery*.

La valeur de seuil par défaut du récepteur de journal est de 100 000 Ko. Vous pouvez définir une valeur plus élevée lorsque vous créez le gestionnaire de files d'attente. La valeur initiale de l'attribut LogReceiverSize est écrite dans la section LogDefaults du fichier mqs.ini.

Lorsqu'un récepteur de journal dépasse le seuil indiqué, il est détaché et un nouveau récepteur de journal est créé, héritant des attributs du récepteur précédent. Les modifications apportées aux attributs LogReceiverSize ou LogASP après la création d'un gestionnaire de files d'attente sont ignorées lorsque le système associe automatiquement un nouveau récepteur de journal.

Pour plus d'informations sur la configuration du système, voir [Modification des informations de configuration IBM MQ sur Multiplatforms](#).

Si vous devez modifier la taille des récepteurs de journal après la création du gestionnaire de files d'attente, créez un nouveau récepteur de journal et définissez son propriétaire sur QMQM à l'aide des commandes suivantes:

```
CRTJRNRCV JRNRCV(QM GRLIB/AMQ Arnnnnn) THRESHOLD(xxxxxx) +  
TEXT('MQM LOCAL JOURNAL RECEIVER')  
CHGOBJOWN OBJ(QM GRLIB/AMQ Arnnnnn) OBJTYPE(*JRNRCV) NEWOWN(QMQM)
```

Où

### **QMGLIB**

Est le nom de la bibliothèque de votre gestionnaire de files d'attente

### **A**

Est l'identificateur d'instance (généralement A).

### **rnrrrrr**

Récepteur de journal suivant dans la séquence d'appellation décrite précédemment

### **xxxxxx**

Est le nouveau seuil de récepteur (en ko)

**Remarque :** La taille maximale du récepteur est régie par le système d'exploitation. Pour vérifier cette valeur, consultez le mot clé THRESHOLD de la commande **CRTJRNRCV** .

A présent, attachez le nouveau récepteur au journal AMQAJRN à l'aide de la commande suivante:

```
CHGJRN JRN(QMGLIB/AMQ A JRN) JRNRCV(QMGLIB/AMQ Annnnnn)
```

Pour plus de détails sur la gestion de ces récepteurs de journal, voir [«Gestion des journaux sous IBM i»](#), à la page 425 .

## **IBM i** *Utilisation du journal du gestionnaire de files d'attente sous IBM i*

Utilisez ces informations pour comprendre comment IBM MQ for IBM i utilise les journaux dans son fonctionnement pour contrôler les mises à jour des objets locaux.

Les mises à jour persistantes des files d'attente de messages sont effectuées en deux étapes. Les enregistrements représentant la mise à jour sont d'abord écrits dans le journal, puis le fichier de file d'attente est mis à jour.

Les récepteurs de journal peuvent donc devenir plus à jour que les fichiers de file d'attente. Pour s'assurer que le processus de redémarrage commence à partir d'un point cohérent, IBM MQ utilise des points de contrôle.

Un point de contrôle est un point dans le temps où l'enregistrement décrit dans le journal est identique à l'enregistrement dans la file d'attente. Le point de contrôle lui-même est constitué de la série d'enregistrements de journal nécessaires pour redémarrer le gestionnaire de files d'attente. Par exemple, l'état de toutes les transactions (c'est-à-dire les unités de travail) actives au moment du point de contrôle.

Les points de contrôle sont générés automatiquement par IBM MQ. Ils sont utilisés lorsque le gestionnaire de files d'attente démarre et s'arrête, et après un certain nombre d'opérations consignées.

Vous pouvez forcer un gestionnaire de files d'attente à prendre un point de contrôle en émettant la commande RCDMQMIMG sur tous les objets d'un gestionnaire de files d'attente et en affichant les résultats, comme suit:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNTA(*YES)
```

Lorsque les files d'attente traitent d'autres messages, l'enregistrement de point de contrôle devient incohérent avec l'état en cours des files d'attente.

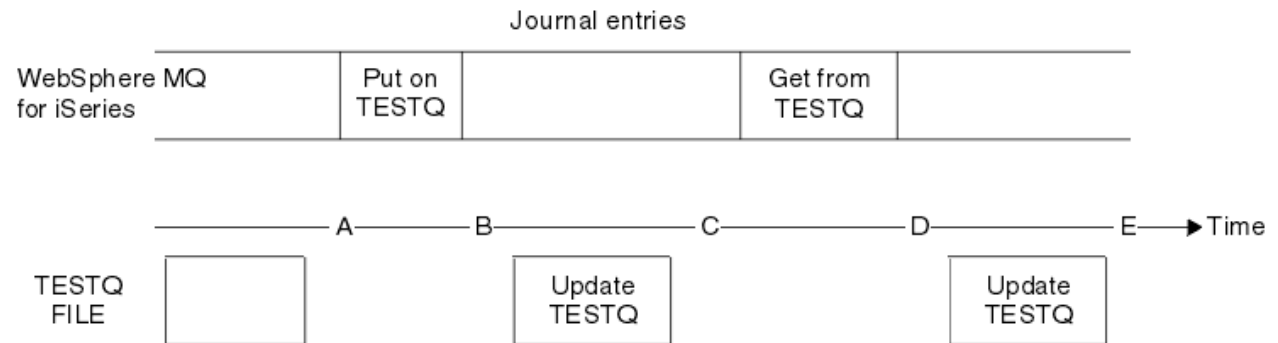
Lorsque IBM MQ est redémarré, il localise le dernier enregistrement de point de contrôle dans le journal. Ces informations sont contenues dans le fichier de point de contrôle qui est mis à jour à la fin de chaque

point de contrôle. L'enregistrement de point de contrôle représente le point de cohérence le plus récent entre le journal et les données. Les données de ce point de contrôle sont utilisées pour régénérer les files d'attente telles qu'elles existaient au moment du point de contrôle. Lorsque les files d'attente sont recrées, le journal est alors lu pour ramener les files d'attente à l'état dans lequel elles se trouvaient avant l'échec du système ou la fermeture.

Pour comprendre comment IBM MQ utilise le journal, prenez en compte le cas d'une file d'attente locale appelée TESTQ dans le gestionnaire de files d'attente TEST. Il est représenté par le fichier IFS:

```
/QIBM/UserData/mqm/qmgrs/TEST/queues
```

Si un message spécifié est inséré dans cette file d'attente, puis extrait de la file d'attente, les actions qui ont lieu sont présentées dans la figure [Figure 22, à la page 422](#).



*Figure 22. Séquence d'événements lors de la mise à jour des objets MQM*

Les cinq points, de A à E, représentés dans le diagramme représentent des points dans le temps qui définissent les états suivants:

- A** La représentation du fichier IFS de la file d'attente est cohérente avec les informations contenues dans le journal.
- B** Un poste de journal est écrit dans le journal définissant une opération Put dans la file d'attente.
- C** La mise à jour appropriée est effectuée dans la file d'attente.
- D** Un poste de journal est écrit dans le journal définissant une opération Get à partir de la file d'attente.
- E** La mise à jour appropriée est effectuée dans la file d'attente.

La clé des fonctions de reprise de IBM MQ for IBM i est que l'utilisateur peut sauvegarder la représentation de fichier IFS de TESTQ comme à l'heure A, puis récupérer la représentation de fichier IFS de TESTQ comme à l'heure E, en restaurant l'objet sauvegardé et en réalisant les entrées dans le journal à partir de l'heure A .

Cette stratégie est utilisée par IBM MQ for IBM i pour récupérer les messages persistants après une défaillance du système. IBM MQ se souvient d'une entrée particulière dans les récepteurs de journal et s'assure qu'au démarrage, il réexécute les entrées dans les journaux à partir de ce point. Cette entrée de démarrage est régulièrement recalculée afin que IBM MQ n'ait à effectuer que la relecture minimale nécessaire au prochain démarrage.

IBM MQ permet la récupération individuelle des objets. Toutes les informations persistantes relatives à un objet sont enregistrées dans les journaux IBM MQ for IBM i locaux. Tout objet IBM MQ endommagé ou endommagé peut être entièrement régénéré à partir des informations contenues dans le journal.

Pour plus d'informations sur la façon dont le système gère les récepteurs, voir [«Disponibilité, sauvegarde, récupération et redémarrage sur IBM i»](#), à la page 419.

## IBM i **Images de support sous IBM i**

Sous IBM i, une image de support est une copie complète d'un objet IBM MQ qui est enregistré dans le journal. Certains objets endommagés ou endommagés peuvent être récupérés automatiquement à partir de leur image de support.

Un objet IBM MQ de longue durée peut représenter un grand nombre d'entrées de journal, remontant au point où il a été créé. Pour éviter cela, IBM MQ for IBM i a le concept d'image de support d'un objet.

Cette image de support est une copie complète de l'objet IBM MQ enregistré dans le journal. Si une image d'un objet est prise, l'objet peut être régénéré en réalisant les postes de journal à partir de cette image. Le poste du journal qui représente le point de relecture pour chaque objet IBM MQ est appelé poste de reprise sur incident lié au support. IBM MQ effectue le suivi des éléments suivants:

- Poste de reprise sur incident lié au support pour chaque objet gestionnaire de files d'attente.
- Entrée la plus ancienne de cet ensemble (voir le message d'erreur AMQ7462 dans [«Gestion des journaux sous IBM i»](#), à la page 425 pour plus de détails).

Les images de l'objet \*CTLG et de l'objet \*MQM sont prises régulièrement car ces objets sont essentiels au redémarrage du gestionnaire de files d'attente.

Des images d'autres objets sont prises lorsque cela vous convient. Par défaut, les images de tous les objets sont prises lorsqu'un gestionnaire de files d'attente est arrêté à l'aide de la commande **ENDMQM** avec le paramètre ENDCCTJOB (\*YES). Cette opération peut prendre beaucoup de temps pour les gestionnaires de files d'attente de très grande taille. Si vous devez vous arrêter rapidement, indiquez le paramètre RCDMQMIMG (\*NO) avec ENDCCTJOB (\*YES). Dans ce cas, il est recommandé d'enregistrer une image de support complète dans les journaux après le redémarrage du gestionnaire de files d'attente, à l'aide de la commande suivante:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME)
```

IBM MQ enregistre automatiquement une image d'un objet, s'il trouve un point pratique où un objet peut être décrit de manière compacte par une petite entrée dans le journal. Toutefois, cela peut ne jamais se produire pour certains objets, par exemple, les files d'attente qui contiennent systématiquement un grand nombre de messages.

Plutôt que d'autoriser la date de l'entrée de reprise sur support la plus ancienne à se poursuivre pendant une période inutilement longue, utilisez la IBM MQ commande RCDMQMIMG, qui vous permet de prendre manuellement une image des objets sélectionnés.

## **Reprise à partir d'images de support**

IBM MQ récupère automatiquement certains objets à partir de leur image de support s'il s'avère qu'ils sont endommagés ou endommagés. En particulier, cela s'applique aux objets spéciaux \*MQM et \*CTLG dans le cadre du démarrage normal du gestionnaire de files d'attente. Si une transaction de point de synchronisation était incomplète au moment du dernier arrêt du gestionnaire de files d'attente, toute file d'attente affectée est également récupérée automatiquement afin de terminer l'opération de démarrage.

Vous devez récupérer d'autres objets manuellement à l'aide de la commande IBM MQ RCRMQOBJ. Cette commande réexécute les entrées du journal pour recréer l'objet IBM MQ. Si un objet IBM MQ est endommagé, les seules actions valides sont de le supprimer ou de le recréer par cette méthode. Notez toutefois que les messages non persistants ne peuvent pas être récupérés de cette manière.

## IBM i **Points de contrôle sur IBM MQ for IBM i**

Des points de contrôle sont pris à différents moments pour fournir un point de départ cohérent connu pour la reprise.

L'unité d'exécution de point de contrôle dans le processus AMQZMUC0 est chargée de prendre le point de contrôle aux points suivants:

- Démarrage du gestionnaire de files d'attente (STRMQM).

- Arrêt du gestionnaire de files d'attente (ENDMQM).
- Après un certain temps écoulé depuis le dernier point de contrôle (la période par défaut est de 30 minutes) et un nombre minimum d'enregistrements de journal ont été écrits depuis le point de contrôle précédent (la valeur par défaut est 100).
- Après l'écriture d'un certain nombre d'enregistrements de journal. La valeur par défaut est 10 000.
- Une fois que la taille de seuil du journal a été dépassée, un nouveau récepteur de journal a été créé automatiquement.
- Lorsqu'une image de support complète est prise avec:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(Q_MGR_NAME) DSPJRNDTA(*YES)
```

**IBM i**

## Sauvegardes des données IBM MQ for IBM i

Utilisez ces informations pour comprendre les deux types de sauvegarde IBM MQ pour chaque gestionnaire de files d'attente.

Pour chaque gestionnaire de files d'attente, il existe deux types de sauvegarde IBM MQ à prendre en compte:

- Sauvegarde des données et du journal.

Pour vous assurer que les deux ensembles de données sont cohérents, effectuez cette opération uniquement après avoir arrêté le gestionnaire de files d'attente.

- Sauvegarde du journal.

Vous pouvez effectuer cette opération lorsque le gestionnaire de files d'attente est actif.

Pour les deux méthodes, vous devez rechercher les noms du répertoire IFS du gestionnaire de files d'attente et de la bibliothèque du gestionnaire de files d'attente. Vous pouvez les trouver dans le fichier de configuration IBM MQ (mq.ini). Pour plus d'informations, voir [Modification des informations de configuration IBM MQ sur Multiplatforms](#).

Utilisez les procédures suivantes pour effectuer les deux types de sauvegarde:

### Sauvegarde des données et du journal d'un gestionnaire de files d'attente particulier

**Remarque : N'utilisez pas de demande de sauvegarde en mise à jour lorsque le gestionnaire de files d'attente est en cours d'exécution. Une telle demande ne peut aboutir que si toutes les définitions de validation comportant des modifications en attente sont validées ou invalidées. Si cette commande est utilisée lorsque le gestionnaire de files d'attente est actif, les connexions de canal risquent de ne pas s'arrêter normalement. Utilisez toujours la procédure suivante.**

1. Créez un récepteur de journal vide à l'aide de la commande suivante:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Utilisez la commande **RCDMQMIMG** pour enregistrer une image MQM pour tous les objets IBM MQ , puis forcez un point de contrôle à l'aide de la commande:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Arrêtez les canaux et vérifiez que le gestionnaire de files d'attente n'est pas en cours d'exécution. Si votre gestionnaire de files d'attente est en cours d'exécution, arrêtez-le à l'aide de la commande **ENDMQM** .
4. Sauvegardez la bibliothèque du gestionnaire de files d'attente en exécutant la commande suivante:

```
SAVLIB LIB(QMTEST)
```



5. Sauvegardez les répertoires IFS du gestionnaire de files d'attente en exécutant la commande suivante:

```
SAV DEV(...) OBJ((' /QIBM/UserData/mqm/qmgrs/test'))
```

### Sauvegarde de journal d'un gestionnaire de files d'attente particulier

Etant donné que toutes les informations pertinentes sont conservées dans les journaux, tant que vous effectuez une sauvegarde intégrale à un moment donné, des sauvegardes partielles peuvent être effectuées en sauvegardant les récepteurs de journal. Elles enregistrent toutes les modifications apportées depuis l'heure de la sauvegarde intégrale et sont exécutées à l'aide des commandes suivantes:

1. Créez un récepteur de journal vide à l'aide de la commande suivante:

```
CHGJRN JRN(QMTEST/AMQAJRN) JRNRCV(*GEN)
```

2. Utilisez la commande **RCDMQMIMG** pour enregistrer une image MQM pour tous les objets IBM MQ , puis forcez un point de contrôle à l'aide de la commande:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) DSPJRNDTA(*YES) MQMNAME(TEST)
```

3. Sauvegardez les récepteurs de journal à l'aide de la commande suivante:

```
SAVOBJ OBJ(AMQ*) LIB(QMTEST) OBJTYPE(*JRNRCV) .....
```

Une stratégie de sauvegarde simple consiste à effectuer une sauvegarde intégrale des bibliothèques IBM MQ toutes les semaines et à effectuer une sauvegarde quotidienne du journal. Cela dépend, bien sûr, de la manière dont vous avez configuré votre stratégie de sauvegarde pour votre entreprise.

## **Gestion des journaux sous IBM i**

Dans le cadre de votre stratégie de sauvegarde, prenez soin de vos récepteurs de journal. Il est utile de supprimer les récepteurs de journal des bibliothèques IBM MQ pour diverses raisons:

- Pour libérer de l'espace ; s'applique à tous les récepteurs de journal
- Pour améliorer les performances lors du démarrage (STRMQM)
- Pour améliorer les performances de la création d'objets (RCRMQMOBJ)

Avant de supprimer un récepteur de journal, vous devez vous assurer que vous disposez d'une copie de sauvegarde et que vous n'en avez plus besoin.

Les récepteurs de journal peuvent être supprimés de la bibliothèque du gestionnaire de files d'attente *après* avoir été déconnectés des journaux et sauvegardés, à condition qu'ils soient disponibles pour être restaurés si nécessaire pour une opération de reprise.

Le concept de gestion de journal est illustré dans la [Figure 23](#), à la page 426.

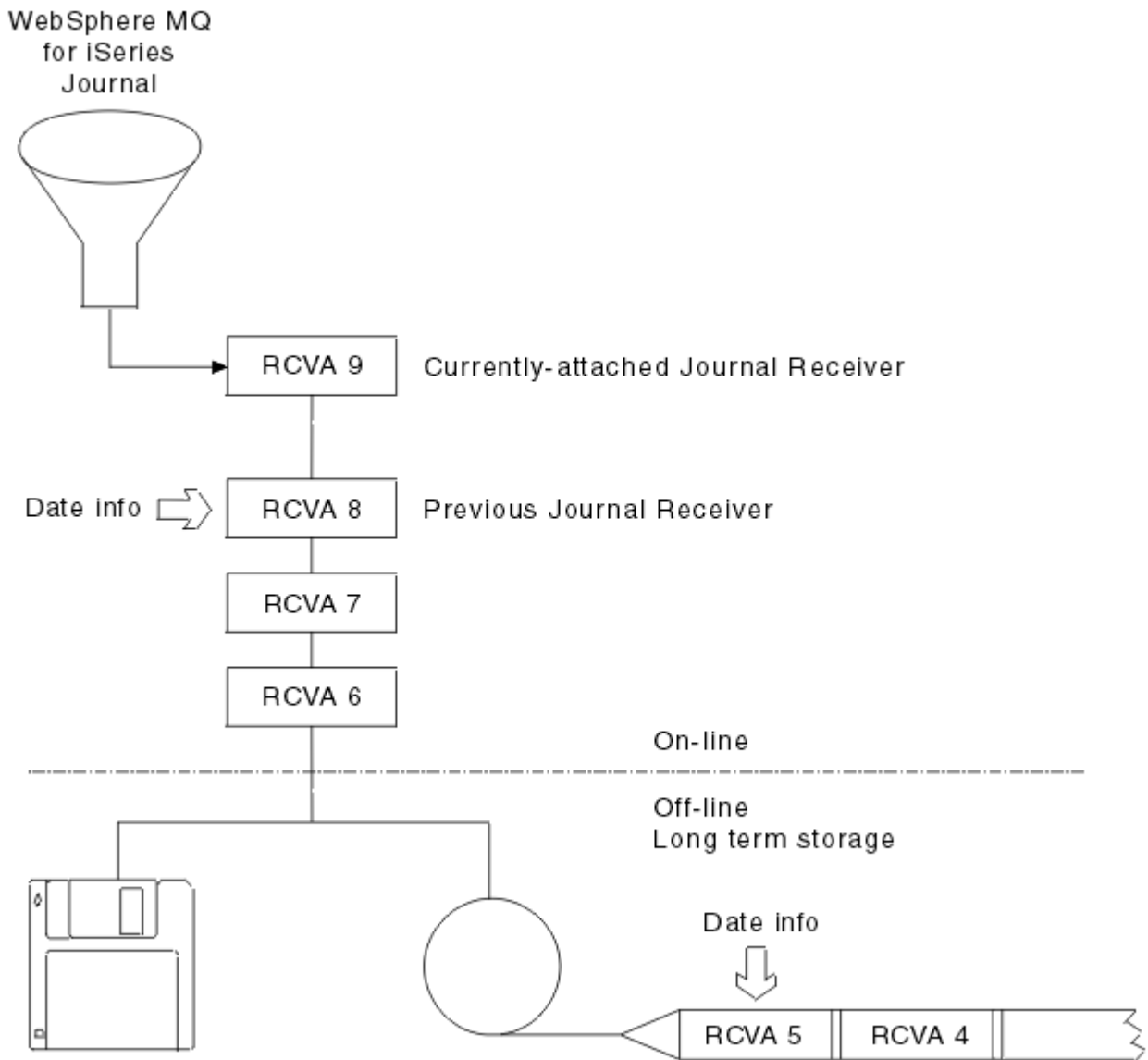


Figure 23. Journalisation sur IBM i

Il est important de savoir jusqu'où se trouvent les journaux IBM MQ , afin de déterminer quand un récepteur de journal qui a été sauvegardé peut être supprimé de la bibliothèque du gestionnaire de files d'attente et quand la sauvegarde elle-même peut être supprimée.

IBM MQ émet deux messages dans la file d'attente de messages du gestionnaire de files d'attente (QMQMMSG dans la bibliothèque du gestionnaire de files d'attente) pour vous aider à déterminer cette heure. Ces messages sont émis lorsqu'il démarre, lorsqu'il modifie un récepteur de journal local et que vous utilisez RCDMQIMG pour forcer un point de contrôle. Les deux messages sont les suivants:

**AMQ7460**

Point de reprise au démarrage. Ce message définit la date et l'heure de l'entrée de démarrage à partir de laquelle IBM MQ réexécute le journal en cas de reprise au démarrage. Si le récepteur de journal contenant cet enregistrement est disponible dans les bibliothèques IBM MQ , ce message contient également le nom du récepteur de journal contenant l'enregistrement.

**AMQ7462**

Poste de reprise de support le plus ancien. Ce message définit la date et l'heure de l'entrée la plus ancienne à utiliser pour recréer un objet à partir de son image de support.

Le récepteur de journal identifié est le plus ancien requis. Les autres récepteurs de journal IBM MQ ayant des dates de création plus anciennes ne sont plus nécessaires. Si seules des étoiles sont

affichées, vous devez restaurer les sauvegardes à partir de la date indiquée pour déterminer le récepteur de journal le plus ancien.

Lorsque ces messages sont consignés, IBM MQ écrit également un objet d'espace utilisateur dans la bibliothèque du gestionnaire de files d'attente qui contient une seule entrée: le nom du récepteur de journal le plus ancien qui doit être conservé sur le système. Cet espace utilisateur est appelé AMQJRNINF et les données sont écrites au format suivant:

```
JJJJJJJJJLLLLLLLLLLLLYYYYMMDDHHMMSSmmm
```

où :

**JJJJJJJJJ**

Est le nom de récepteur le plus ancien dont IBM MQ a toujours besoin.

**LLLLLLLLLL**

Nom de la bibliothèque du récepteur de journal.

**YYYY**

Année de l'entrée de journal la plus ancienne dont IBM MQ a besoin.

**MM**

Mois de l'entrée de journal la plus ancienne dont IBM MQ a besoin.

**DD**

Jour de l'entrée de journal la plus ancienne dont IBM MQ a besoin.

**HH**

Heure de l'entrée de journal la plus ancienne dont IBM MQ a besoin.

**SS**

Nombre de secondes de l'entrée de journal la plus ancienne dont IBM MQ a besoin.

**mmm**

Nombre de millisecondes de l'entrée de journal la plus ancienne dont IBM MQ a besoin.

Lorsque le récepteur de journal le plus ancien a été supprimé du système, cet espace utilisateur contient des astérisques (\*) pour le nom du récepteur de journal.

**Remarque :** L'exécution périodique de RCDMQMIMG OBJ(\*ALL) OBJTYPE(\*ALL) DSPJRNDTA(\*YES) permet d'économiser du temps de démarrage pour IBM MQ et de réduire le nombre de récepteurs de journal locaux dont vous avez besoin pour la sauvegarde et la restauration en vue de la reprise.

IBM MQ for IBM i ne fait référence aux récepteurs de journal que s'il effectue un processus de reprise soit pour le démarrage, soit pour la recréation d'un objet. S'il détecte qu'un journal dont il a besoin n'est pas présent, il émet le message AMQ7432 dans la file d'attente de messages du gestionnaire de files d'attente (QMOMMSG), en indiquant l'heure et la date de l'entrée de journal dont il a besoin pour terminer le processus de reprise.

Si tel est le cas, restaurez tous les récepteurs de journal qui ont été déconnectés après cette date à partir de la sauvegarde, afin de permettre la réussite du processus de reprise.

Conservez le récepteur de journal qui contient le poste de démarrage, ainsi que les récepteurs de journal suivants, disponibles dans la bibliothèque du gestionnaire de files d'attente.

Conservez le récepteur de journal contenant le plus ancien Media Recovery Entry, ainsi que tous les récepteurs de journal suivants, disponibles à tout moment et présents dans la bibliothèque du gestionnaire de files d'attente ou sauvegardés.

Lorsque vous forcez un point de contrôle:

- Si le récepteur de journal nommé dans AMQ7460 n'est pas avancé, cela indique qu'une unité de travail incomplète doit être validée ou annulée.
- Si le récepteur de journal nommé dans AMQ7462 n'est pas avancé, cela indique qu'il existe un ou plusieurs objets endommagés.

## **IBM i** *Restauration d'un gestionnaire de files d'attente complet (données et journaux) sous IBM i*

Utilisez ces informations pour restaurer un ou plusieurs gestionnaires de files d'attente à partir d'une sauvegarde ou d'une machine distante.

Si vous devez récupérer un ou plusieurs gestionnaires de files d'attente IBM MQ à partir d'une sauvegarde, procédez comme suit.

1. Mettez au repos les gestionnaires de files d'attente IBM MQ .
2. Localisez votre dernier groupe de sauvegarde, constitué de votre sauvegarde intégrale la plus récente et des récepteurs de journal sauvegardés par la suite.
3. Effectuez une opération RSTLIB, à partir de la sauvegarde intégrale, pour restaurer les bibliothèques de données IBM MQ à leur état au moment de la sauvegarde intégrale, en exécutant les commandes suivantes:

```
RSTLIB LIB(QMQLIB1) .....  
RSTLIB LIB(QMQLIB2) .....
```

Si un récepteur de journal a été partiellement sauvegardé dans une sauvegarde de journal et entièrement sauvegardé dans une sauvegarde ultérieure, restaurez uniquement la sauvegarde complète. Restaurez les journaux individuellement, dans l'ordre chronologique.

4. Effectuez une opération RST pour la restauration des répertoires IFS IBM MQ sur le système de fichiers IFS, à l'aide de la commande suivante:

```
RST DEV(...) OBJ((' /QIBM/UserData/mqm/qmgrs/testqm')) ...
```

5. Démarrez le gestionnaire de files d'attente de messages. Cette commande réexécute tous les enregistrements de journal écrits depuis la sauvegarde intégrale et restaure tous les objets IBM MQ à l'état cohérent au moment de la sauvegarde du journal.

Si vous souhaitez restaurer un gestionnaire de files d'attente complet sur une machine différente, utilisez la procédure suivante pour tout restaurer à partir de la bibliothèque du gestionnaire de files d'attente. (Nous utilisons TEST comme exemple de nom de gestionnaire de files d'attente.)

1. CRTMQM TEST
2. DLTLIB LIB(QMTEST)
3. RSTLIB SAVLIB(QMTEST) DEV(\*SAVF) SAVF(QMGRLIBSAV)
4. Supprimez les fichiers IFS suivants:

```
/QIBM/UserData/mqm/qmgrs/TEST/QMQMCHKPT  
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMQMOBJCAT  
/QIBM/UserData/mqm/qmgrs/TEST/qmanager/QMANAGER  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.AUTH.DATA.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CHANNEL.INITQ/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.COMMAND.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.REPOSITORY.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.CLUSTER.TRANSMIT.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.PENDING.DATA.QUEUE/q  
/QIBM/UserData/mqm/qmgrs/TEST/queues/SYSTEM.ADMIN.COMMAND.QUEUE/q
```

5. STRMQM TEST
6. RCRMQMOBJ OBJ(\*ALL) OBJTYPE(\*ALL) MQMNAME(TEST)

## **IBM i** *Restauration de récepteurs de journal pour un gestionnaire de files d'attente particulier sous IBM i*

Utilisez ces informations pour comprendre les différentes manières de restaurer les récepteurs de journal.

L'action la plus courante consiste à restaurer un récepteur de journal sauvegardé dans une bibliothèque de gestionnaire de files d'attente, si un récepteur qui a été supprimé est à nouveau nécessaire pour une fonction de reprise ultérieure.

Il s'agit d'une tâche simple qui nécessite la restauration des récepteurs de journal à l'aide de la commande IBM i RSTOBJ standard:

```
RSTOBJ OBJ(QMQMDATA/AMQA000005) OBJTYPE(*JRNRCV) .....
```

Il peut être nécessaire de restaurer une série de récepteurs de journal plutôt qu'un seul. Par exemple, AMQA000007 est le récepteur le plus ancien dans les bibliothèques IBM MQ et AMQA000005 et AMQA000006 doivent être restaurés.

Dans ce cas, restaurez les récepteurs individuellement dans l'ordre chronologique inverse. Ce n'est pas toujours nécessaire, mais c'est une bonne pratique. Dans les situations graves, vous devrez peut-être utiliser la IBM i commande WRKJRNA pour associer les récepteurs de journal restaurés au journal.

Lors de la restauration de journaux, le système crée automatiquement un récepteur de journal attaché avec un nouveau nom dans la séquence de récepteurs de journal. Toutefois, le nouveau nom généré peut être identique à celui d'un récepteur de journal que vous devez restaurer. Une intervention manuelle est nécessaire pour résoudre ce problème ; pour créer un nouveau récepteur de journal de nom dans l'ordre et un nouveau journal avant de restaurer le récepteur de journal.

Par exemple, prenez en compte l'incident lié au journal sauvegardé AMQAJRN et aux récepteurs de journal suivants:

- AMQA000000
- AMQA100000
- AMQA200000
- AMQA300000
- AMQA400000
- AMQA500000
- AMQA600000
- AMQA700000
- AMQA800000
- AMQA900000

Lors de la restauration du journal AMQAJRN dans une bibliothèque de gestionnaire de files d'attente, le système crée automatiquement le récepteur de journal AMQA000000. Ce récepteur généré automatiquement entre en conflit avec l'un des récepteurs de journal existants (AMQA000000) que vous souhaitez restaurer et que vous ne pouvez pas restaurer.

La solution est la suivante :

1. Créez manuellement le récepteur de journal suivant (voir [«Journaux de gestionnaire de files d'attente sous IBM i»](#), à la page 419 ):

```
CRTJRNRCV JRNRCV(QMQRLIB/AMQA900001) THRESHOLD(XXXXX)
```

2. Créez manuellement le journal avec le récepteur de journal:

```
CRTJRN JRN(QMGLIB/AMQAJRN) MNGRCV(*SYSTEM) +  
JRNRVC(QMGLIB/AMQA9000001) MSGQ(QMGLIB/AMQAJRNMSG)
```

3. Restaurez les récepteurs de journal locaux AMQA000000 sur AMQA900000.

IBM i

## Gestionnaires de files d'attente multi-instances sous IBM i

Les gestionnaires de files d'attente multi-instances permettent d'améliorer la disponibilité du système en basculant automatiquement vers un serveur de secours en cas de défaillance du serveur actif. Les serveurs actifs et de secours hébergent plusieurs instances d'un même gestionnaire de files d'attente et partagent des données identiques le concernant. Si l'instance active échoue, vous devez transférer son journal vers la base de données de secours qui prend le relais afin que le gestionnaire de files d'attente puisse régénérer ses files d'attente.

Configurez les systèmes IBM i sur lesquels vous exécutez des gestionnaires de files d'attente multi-instance de sorte que, si l'instance de gestionnaire de files d'attente active échoue, le journal qu'elle utilise soit disponible pour l'instance de secours qui prend le relais. Vous pouvez concevoir vos propres tâches de configuration et d'administration pour rendre le journal de l'instance active disponible à l'instance qui prend le relais. Si vous ne souhaitez pas perdre de messages, votre conception doit vous assurer que le journal de secours est cohérent avec le journal actif au point de défaillance. Vous pouvez adapter votre conception à partir de l'une des deux configurations décrites avec des exemples dans les rubriques suivantes qui assurent la cohérence.

1. Mettez en miroir le journal du système qui exécute l'instance de gestionnaire de files d'attente active vers les systèmes qui exécutent des instances de secours.
2. Placez le journal dans un pool de mémoire secondaire indépendant (IASP) qui est transférable du système exécutant l'instance active vers une instance de secours.

La première solution ne nécessite aucun matériel ou logiciel supplémentaire car elle utilise des ASP de base. La deuxième solution nécessite des ASP indépendants commutables qui ont besoin d'une prise en charge de la mise en cluster IBM i disponible sous la forme d'un IBM i produit de licence 5761-SS1 à prix séparé.

IBM i

## Fiabilité et disponibilité sous IBM i

Les gestionnaires de files d'attente multi-instance visent à améliorer la disponibilité des applications. Les contraintes technologiques et physiques signifient que vous avez besoin de solutions différentes pour répondre aux exigences de la reprise après incident, de la sauvegarde des gestionnaires de files d'attente et de l'exploitation en continu.

Lors de la configuration de la fiabilité et de la disponibilité, vous échangez un grand nombre de facteurs, ce qui se traduit par quatre points de conception distincts:

### Reprise après incident

Optimisé pour la reprise après un sinistre majeur qui détruit tous vos actifs locaux.

La reprise après incident sous IBM i est souvent basée sur la protection géographique par disque miroir de l'ASP indépendant.

### Sauvegarde

Optimisé pour la récupération après une panne localisée, généralement une erreur humaine ou un problème technique imprévu.

IBM MQ fournit des gestionnaires de files d'attente de sauvegarde pour la sauvegarde périodique des gestionnaires de files d'attente. Vous pouvez également utiliser la réplication asynchrone des journaux de gestionnaire de files d'attente pour améliorer la mise à jour de la sauvegarde.

### Disponibilité

Optimisé pour la restauration rapide des opérations donnant l'apparence d'un service quasi ininterrompu suite à des défaillances techniques prévisibles telles qu'une panne de serveur ou de disque.

La récupération est généralement mesurée en minutes, la détection prenant parfois plus de temps que le processus de récupération. Un gestionnaire de files d'attente multi-instance vous aide à configurer la *disponibilité*.

### **Fonctionnement en continu**

Optimisé pour fournir un service ininterrompu.

Les solutions de fonctionnement continu doivent résoudre le problème de détection, et impliquent presque toujours de soumettre le même travail à travers plus d'un système et soit en utilisant le premier résultat, soit si l'exactitude est une considération majeure, en comparant au moins deux résultats.

Un gestionnaire de files d'attente multi-instance vous aide à configurer la *disponibilité*. Une instance du gestionnaire de files d'attente est active à la fois. Le basculement vers une instance de secours prend entre un peu plus de dix secondes et quinze minutes ou plus, selon la manière dont le système est configuré, chargé et réglé.

Un gestionnaire de files d'attente multi-instance peut donner l'apparence d'un service presque ininterrompu s'il est utilisé avec des IBM MQ MQI clients reconnectables, qui peuvent continuer le traitement sans que le programme d'application ait nécessairement connaissance d'une indisponibilité du gestionnaire de files d'attente ; voir la rubrique [Reconnexion client automatisée](#).



### **Composants d'une solution à haute disponibilité sur IBM i**

Construisez une solution à haute disponibilité à l'aide de gestionnaires de files d'attente multi-instance en fournissant un stockage en réseau robuste pour les données de gestionnaire de files d'attente, une réplication de journal ou un stockage IASP robuste pour les journaux de gestionnaire de files d'attente, et en utilisant des clients reconnectables, d'applications configurées en tant que services de gestionnaire de files d'attente redémarrables.

Un gestionnaire de files d'attente multi-instance réagit à la détection de l'échec du gestionnaire de files d'attente en reprenant le démarrage d'une autre instance de gestionnaire de files d'attente sur un autre serveur. Pour terminer son démarrage, l'instance doit accéder aux données du gestionnaire de files d'attente partagées dans le stockage en réseau et à sa copie du journal du gestionnaire de files d'attente local.

Pour créer une solution à haute disponibilité, vous devez gérer la disponibilité des données du gestionnaire de files d'attente, la mise à jour du journal du gestionnaire de files d'attente local et générer des applications client reconnectables ou déployer vos applications en tant que services de gestionnaire de files d'attente pour redémarrer automatiquement lorsque le gestionnaire de files d'attente reprend. La reconnexion client automatique n'est pas prise en charge par IBM MQ classes for Java.

### **Données de gestionnaire de files d'attente**

Placez les données du gestionnaire de files d'attente dans un stockage en réseau partagé, hautement disponible et fiable, éventuellement à l'aide de disques RAID de niveau 1 ou supérieur. Le système de fichiers doit répondre aux exigences d'un système de fichiers partagé pour les gestionnaires de files d'attente multi-instance. Pour plus d'informations sur les exigences pour les systèmes de fichiers partagés, voir [Exigences pour les systèmes de fichiers partagés](#). Network File System 4 (NFS4) est un protocole qui répond à ces exigences.

### **Journaux du gestionnaire de files d'attente**

Vous devez également configurer les journaux IBM i utilisés par les instances de gestionnaire de files d'attente afin que l'instance de secours puisse restaurer les données de son gestionnaire de files d'attente à un état cohérent. Pour un service ininterrompu, cela signifie que vous devez restaurer les journaux à leur état lorsque l'instance active a échoué. Contrairement aux solutions de sauvegarde ou de reprise après incident, la restauration des journaux à un point de contrôle antérieur n'est pas suffisante.

Vous ne pouvez pas partager physiquement des journaux entre plusieurs systèmes IBM i sur le stockage en réseau. Pour restaurer les journaux de gestionnaire de files d'attente à l'état cohérent au point de défaillance, vous devez soit transférer le journal physique qui était local à l'instance de gestionnaire de

files d'attente active au moment de la défaillance à la nouvelle instance qui a été activée, soit conserver des miroirs du journal sur les instances de secours en cours d'exécution. Le journal en miroir est une réplique de journal éloigné qui a été maintenue exactement en synchronisation avec le journal local appartenant à l'instance ayant échoué.

Trois configurations sont des points de départ pour la conception de la gestion des journaux d'un gestionnaire de files d'attente multi-instance.


1. Utilisation de la réplique de journal synchronisée (protection par disque miroir) à partir de l'ASP de l'instance active vers les ASP des instances de secours.
2. Transfert d'un ASP indépendant que vous avez configuré pour stocker le journal du gestionnaire de files d'attente de l'instance active vers l'instance de secours qui prend le relais en tant qu'instance active.
3. Utilisation de miroirs d'ASP indépendant secondaires synchronisés.

Pour plus d'informations sur l'insertion de données de gestionnaire de files d'attente dans un iASP, voir les options [ASP](#) dans la commande IBM MQ IBM i CRTMQM.

Voir aussi [Haute disponibilité](#) dans les informations IBM i dans IBM Documentation.

## Applications

Pour générer un client afin de se reconnecter automatiquement au gestionnaire de files d'attente lors de la reprise du gestionnaire de files d'attente de secours, connectez votre application au gestionnaire de files d'attente à l'aide de MQCONNX et spécifiez MQCNO\_RECONNECT\_Q\_MGR dans la zone Options de **MQCNO**. Voir [High availability sample programs](#) pour trois exemples de programmes utilisant des clients reconnectables et [Application recovery](#) pour plus d'informations sur la conception d'applications client pour la reprise.

 *Création d'un partage de réseau pour les données de gestionnaire de files d'attente à l'aide de NetServer sous IBM i*

Créez un partage de réseau sur un serveur IBM i pour stocker les données du gestionnaire de files d'attente. Configurez des connexions à partir de deux serveurs, qui vont héberger des instances de gestionnaire de files d'attente, pour accéder au partage de réseau.

## Avant de commencer

- Vous avez besoin de trois serveurs IBM i pour cette tâche. Le partage de réseau est défini sur l'un des serveurs, GAMMA. Les deux autres serveurs, ALPHA et BETA, doivent se connecter à GAMMA.
- Installez IBM MQ sur les trois serveurs.
- Installez System i Navigator; voir [System i Navigator](#).

## Pourquoi et quand exécuter cette tâche

- Créez le répertoire du gestionnaire de files d'attente sur GAMMA et définissez la propriété et les droits appropriés pour les profils utilisateur QMQM et QMQMADM. Le répertoire et les droits d'accès sont facilement créés en installant IBM MQ sur GAMMA.
- Utilisez System i Navigator pour créer un partage dans le répertoire de données du gestionnaire de files d'attente sur GAMMA.
- Créez des répertoires sur ALPHA et BETA qui pointent vers le partage.

## Procédure

1. Sous GAMMA, créez le répertoire pour héberger les données du gestionnaire de files d'attente avec le profil utilisateur QMQM comme propriétaire et QMQMADM comme groupe principal.

**Conseil :**



Un moyen rapide et fiable de créer le répertoire avec les droits appropriés consiste à installer IBM MQ sur GAMMA.

Par la suite, si vous ne souhaitez pas exécuter IBM MQ sur GAMMA, désinstallez IBM MQ. Après la désinstallation, le répertoire /QIBM/UserData/mqm/qmgrs reste sur GAMMA avec le profil utilisateur QMQM propriétaire et QMQMADM le groupe principal.

La tâche utilise le répertoire /QIBM/UserData/mqm/qmgrs sur GAMMA pour le partage.

2. Démarrez l'assistant System i Navigator **Ajouter une connexion** et connectez-vous au système GAMMA.
  - a) Cliquez deux fois sur l'icône **System i Navigator** sur votre bureau Windows .
  - b) Cliquez sur **Oui** pour créer une connexion.
  - c) Suivez les instructions de l'assistant **Ajouter une connexion** et créez une connexion entre le système IBM i et GAMMA.

La connexion à GAMMA est ajoutée à **Mes connexions**.

3. Ajoutez un nouveau partage de fichiers sur GAMMA.
  - a) Dans la fenêtre **System i Navigator** , cliquez sur le dossier File Shares dans My Connections/GAMMA/File Systems.
  - b) Dans la fenêtre **Mes tâches** , cliquez sur **Gérer les partages IBM i NetServer**.

Une nouvelle fenêtre, **IBM i NetServer -GAMMA**, s'ouvre sur votre bureau et affiche les objets partagés.
  - c) Cliquez avec le bouton droit de la souris sur le dossier Shared Objects > **Fichier** > **Nouveau** > **Fichier**.

Une nouvelle fenêtre, **IBM i NetServer File Share-GAMMA**, s'ouvre.
  - d) Attribuez un nom au partage, WMQ par exemple.
  - e) Définissez le contrôle d'accès sur Read/Write.
  - f) Sélectionnez le **Nom de chemin** en accédant au répertoire /QIBM/UserData/mqm/qmgrs que vous avez créé précédemment, puis cliquez sur **OK**.

La fenêtre **IBM i NetServer File Share-GAMMA** se ferme et WMQ est répertorié dans la fenêtre des objets partagés.

4. Cliquez avec le bouton droit de la souris sur **WMQ** dans la fenêtre des objets partagés. Cliquez sur **Fichier** > **Droits d'accès**.

Une fenêtre s'ouvre, **Qmgrs Permissions-GAMMA**, pour l'objet /QIBM/UserData/mqm/qmgrs.

- a) Vérifiez les droits suivants pour QMQM, s'ils ne sont pas déjà définis:

- Read
- Write
- Execute
- Management
- Existence
- Alter
- Reference

- b) Vérifiez les droits suivants pour QMQMADM, s'ils ne sont pas déjà définis:

- Read
- Write
- Execute
- Reference

- c) Ajoutez d'autres profils utilisateur auxquels vous souhaitez accorder des droits d'accès à /QIBM/UserData/mqm/qmgrs.

Par exemple, vous pouvez accorder au profil utilisateur par défaut (public) Read et Execute les droits d'accès à /QIBM/UserData/mqm/qmgrs.

5. Vérifiez que tous les profils utilisateur autorisés à accéder à /QIBM/UserData/mqm/qmgrs sur GAMMA ont le même mot de passe que sur les serveurs qui accèdent à GAMMA.

En particulier, assurez-vous que les profils utilisateur QMQM sur d'autres serveurs, qui vont accéder au partage, ont le même mot de passe que le profil utilisateur QMQM sur GAMMA.

**Conseil :** Cliquez sur le dossier My Connections/GAMMA/Users and Groups dans le System i Navigateur Navigator pour définir les mots de passe. Vous pouvez également utiliser les commandes **CHFUSRPRF** et **CHGPWD**.

## Résultats

Vérifiez que vous pouvez accéder à GAMMA à partir d'autres serveurs à l'aide du partage. Si vous effectuez les autres tâches, vérifiez que vous pouvez accéder à GAMMA à partir de ALPHA et de BETA en utilisant le chemin /QNTC/GAMMA/WMQ. Si le répertoire /QNTC/GAMMA n'existe pas sur ALPHA ou BETA, vous devez le créer. Selon le domaine NetServer, vous devrez peut-être effectuer un IPL ALPHA ou BETA avant de créer le répertoire.

```
CRTDIR DIR('/QNTC/GAMMA')
```

Une fois que vous avez vérifié que vous avez accès à /QNTC/GAMMA/WMQ à partir de ALPHA ou de BETA, en émettant la commande, `CRTMQM MQMNAME('QM1') MQMDIRP('/QNTC/GAMMA/WMQ')` crée /QIBM/UserData/mqm/qmgrs/QM1 sur GAMMA.

## Que faire ensuite

Créez un gestionnaire de files d'attente multi-instance en suivant les étapes de l'une des tâches, [«Création d'un gestionnaire de files d'attente multi-instance à l'aide de la mise en miroir du journal et de NetServer sous IBM i»](#), à la page 446 ou [«Conversion d'un gestionnaire de files d'attente à instance unique en gestionnaire de files d'attente multi-instance à l'aide de NetServer et de la mise en miroir du journal sous IBM i»](#), à la page 450.

**IBM i**

### **Performances de reprise en ligne sous IBM i**

Le temps nécessaire pour détecter une instance de gestionnaire de files d'attente a échoué, puis pour reprendre le traitement sur une instance de secours peut varier entre des dizaines de secondes et quinze minutes ou plus selon la configuration. Les performances doivent être une considération majeure lors de la conception et du test d'une solution à haute disponibilité.

Il existe des avantages et des inconvénients à prendre en compte lors de la configuration d'un gestionnaire de files d'attente multi-instance pour utiliser la réplication de journal ou un ASP indépendant. La mise en miroir nécessite que le gestionnaire de files d'attente écrive de manière synchrone dans un journal éloigné. D'un point de vue matériel, cela n'a pas besoin d'affecter les performances, mais du point de vue du logiciel, il y a une plus grande longueur de chemin d'accès à l'écriture dans un journal éloigné que dans un journal local, ce qui peut réduire les performances d'un gestionnaire de files d'attente en cours d'exécution dans une certaine mesure. Toutefois, lorsque le gestionnaire de files d'attente de secours prend le relais, le délai de synchronisation de son journal local à partir du journal éloigné géré par l'instance active avant son échec est généralement faible par rapport au temps nécessaire à IBM i pour détecter et transférer l'ASP indépendant sur le serveur exécutant l'instance de secours du gestionnaire de files d'attente. Les temps de transfert de l'ASP indépendant peuvent atteindre jusqu'à dix à quinze minutes au lieu d'être terminés en secondes. Le temps de transfert de l'ASP indépendant dépend du nombre d'objets qui doivent être *mis en fonction* lorsque l'ASP indépendant est transféré sur le système de secours et de la taille des chemins d'accès, ou des index, qui doivent être fusionnés.

Lorsque le gestionnaire de files d'attente de secours prend le relais, le délai de synchronisation de son journal local à partir du journal éloigné géré par l'instance active avant l'échec est généralement faible par rapport au temps nécessaire à IBM i pour détecter et transférer l'ASP indépendant sur le

serveur exécutant l'instance de secours du gestionnaire de files d'attente. Les temps de transfert de l'ASP indépendant peuvent aller jusqu'à dix à quinze minutes au lieu d'être terminés en quelques secondes. La durée de transfert de l'ASP indépendant dépend du nombre d'objets qui doivent être *mis en fonction* lorsque l'ASP indépendant est transféré sur le système de secours et de la taille des chemins d'accès, ou des index, qui doivent être fusionnés.

Toutefois, le transfert du journal n'est pas le seul facteur qui influence le temps nécessaire à la reprise complète de l'instance de secours. Vous devez également prendre en compte le temps nécessaire au système de fichiers du réseau pour libérer le verrou sur les données du gestionnaire de files d'attente qui indiquent à l'instance de secours de tenter de poursuivre son démarrage, ainsi que le temps nécessaire à la reprise des files d'attente à partir du journal afin que l'instance puisse recommencer à traiter les messages. Ces autres sources de retard s'ajoutent au temps nécessaire au démarrage d'une instance de secours. La durée totale du basculement comprend les composants suivants:

#### **Heure de détection des incidents**

Temps nécessaire à NFS pour libérer le verrou sur les données du gestionnaire de files d'attente et à l'instance de secours pour poursuivre son processus de démarrage.

#### **Durée de transfert**

Dans le cas d'un cluster à haute disponibilité, le temps nécessaire à IBM i pour transférer l'ASP indépendant du système hébergeant l'instance active vers l'instance de secours, et dans le cas de la réplication de journal, le temps nécessaire à la mise à jour du journal local sur la base de secours avec les données de la réplique distante.

#### **Heure de redémarrage**

Temps nécessaire à l'instance de gestionnaire de files d'attente nouvellement active pour régénérer ses files d'attente à partir du dernier point de contrôle dans son journal restauré et pour reprendre le traitement des messages.

#### **Remarque :**

Si l'instance de secours qui a pris le relais est configurée pour être répliquée de manière synchrone sur l'instance active précédente, le démarrage peut être retardé. Il se peut que la nouvelle instance activée ne puisse pas être répliquée dans son journal distant, si le journal distant se trouve sur le serveur qui a hébergé l'instance active précédente et que le serveur ait échoué.

Le délai d'attente par défaut d'une réponse synchrone est d'une minute. Vous pouvez configurer le délai maximal avant l'expiration de la réplication. Vous pouvez également configurer des instances de secours pour commencer à utiliser la réplication asynchrone sur l'instance active qui a échoué. Vous basculez ensuite vers la réplication synchrone, lorsque l'instance ayant échoué s'exécute à nouveau sur la base de données de secours. Il en va de même pour l'utilisation de miroirs ASP indépendants synchrones.

Vous pouvez effectuer des mesures de référence distinctes pour ces composants pour vous aider à évaluer la durée globale de la reprise en ligne et à prendre en compte dans votre décision l'approche de configuration à utiliser. Pour prendre la meilleure décision en matière de configuration, vous devez également tenir compte de la manière dont d'autres applications sur le même serveur basculeront et de la présence ou non de processus de sauvegarde ou de reprise après incident qui utilisent déjà l'ASP indépendant.

Les temps de transfert ASP indépendant peuvent être raccourcis en optimisant la configuration de votre grappe:

1. Les profils utilisateur des systèmes du cluster doivent avoir le même ID groupe et le même ID utilisateur pour que le processus de mise en fonction n'ait pas besoin de modifier les ID utilisateur et les ID groupe.
2. Réduisez le nombre d'objets de base de données dans le système et les pools de stockage sur disque utilisateur de base, car ils doivent être fusionnés pour créer la table de références croisées pour le groupe de pools de stockage sur disque.
3. D'autres conseils sur les performances sont disponibles dans le document IBM Redbook, *Implémentation PowerHA for IBM i, SG24-7405*.

Une configuration utilisant des ASP de base, la mise en miroir du journal et une configuration de petite taille doit basculer de l'ordre de plusieurs dizaines de secondes.

## **IBM i** *Présentation de la combinaison des fonctions de mise en cluster IBM i avec la mise en cluster IBM MQ*

L'exécution de IBM MQ sous IBM i et l'exploitation des fonctions de mise en cluster IBM i peuvent fournir une solution à haute disponibilité plus complète que l'utilisation de la mise en cluster IBM MQ uniquement.

Pour disposer de cette fonctionnalité, vous devez configurer:

1. Clusters sur votre machine IBM i ; voir [«Clusters IBM i»](#), à la page 436
2. Un pool de mémoire secondaire indépendant (IASP) dans lequel vous déplacez le gestionnaire de files d'attente ; voir [«Pools de mémoire secondaire indépendants \(IASP\)»](#), à la page 436
3. Un groupe de ressources en grappe (CRG) ; voir [«Groupes de ressources en grappe d'unité»](#), à la page 436, dans lequel vous définissez:
  - Domaine de reprise
  - pool de stockage auxiliaire indépendant
  - Programme d'exit ; voir [«Programme d'exit CRG d'unité»](#), à la page 437

### **Clusters IBM i**

Un cluster IBM i est une collection d'instances, c'est-à-dire d'ordinateurs ou de partitions IBM i , qui sont liées logiquement entre elles.

Le but de ce regroupement est de permettre la sauvegarde de chaque instance, en éliminant un point de défaillance unique et en augmentant la résilience des applications et des données. Avec une grappe créée, les différents types de groupe de ressources en grappe peuvent être configurés pour gérer les applications, les données et les unités de la grappe.

Pour plus d'informations, voir [Création d'une grappe](#) et la commande [CRTCLU \(Créer une grappe\)](#) .

### **Pools de mémoire secondaire indépendants (IASP)**

Un ASP indépendant est un type d'ASP utilisateur qui sert d'extension de l'espace de stockage à un niveau. C'est une pièce de stockage qui, en raison de son indépendance par rapport à la mémoire du système, peut être facilement manipulée sans avoir à effectuer un IPL du système.

Un ASP indépendant peut être facilement basculé vers une autre instance de système d'exploitation ou répliqué vers un ASP indépendant cible sur une autre instance de système d'exploitation. Deux méthodes peuvent être utilisées pour permuter un ASP indépendant entre des instances:

- La première méthode nécessite que tous les ordinateurs de la grappe, et la tour de disque commutable contenant l'ASP indépendant, soient connectés à l'aide d'une boucle HSL (High Speed Link).
- La deuxième méthode requiert que les instances de système d'exploitation soient des partitions sur le même ordinateur IBM i où les processeurs d'entrée-sortie (IOPS) peuvent être commutés entre les partitions. Aucun matériel spécial n'est nécessaire pour pouvoir répliquer un ASP indépendant. La réplication est effectuée à l'aide de TCP/IP sur le réseau.

Pour plus d'informations, voir la commande [Configuration d'ASP d'unité \(CFGDEVASP\)](#) .

### **Groupes de ressources en grappe d'unité**

Il existe plusieurs types de groupes de ressources en grappe (CRG). Pour plus d'informations sur les différents types de groupes de ressources en grappe disponibles, voir [Groupe de ressources en grappe](#).

Cette rubrique se concentre sur un groupe de ressources en grappe d'unité. Un groupe de ressources en grappe d'unité:

- Décrit et gère les ressources d'unité, telles que les pools de mémoire secondaire indépendants (IASP).
- Définit le domaine de reprise des noeuds de grappe
- Affecte un périphérique et
- Affecte le programme d'exit qui traitera les événements de cluster.

Le domaine de reprise indique quel noeud de grappe sera considéré comme le noeud principal. Les autres noeuds sont considérés comme des sauvegardes. Les noeuds de sauvegarde sont également classés dans le domaine de reprise, en spécifiant le noeud qui est la première sauvegarde, la deuxième sauvegarde, etc., en fonction du nombre de noeuds présents dans le domaine de reprise.

En cas de défaillance d'un noeud principal, le programme d'exit est exécuté sur tous les noeuds du domaine de reprise. Le programme d'exit exécuté sur la première sauvegarde peut alors effectuer les initialisations nécessaires pour faire de ce noeud le nouveau noeud principal.

Pour plus d'informations, voir [Création de groupes de ressources en grappe d'unité](#) et la commande [Créer un groupe de ressources en grappe \(CRTCRG\)](#).

## Programme d'exit CRG d'unité

Le service de ressources en grappe du système d'exploitation appelle un programme d'exit de groupe de ressources en grappe d'unité lorsqu'un événement se produit dans l'un des noeuds définis par le domaine de reprise ; par exemple, un événement de basculement ou de commutation.

Un événement de reprise en ligne se produit lorsque le noeud principal de la grappe est défaillant et que les groupes de ressources en grappe sont commutés avec toutes les ressources qu'ils gèrent, et un événement de commutation se produit lorsqu'un groupe de ressources en grappe spécifique est basculé manuellement du noeud principal vers le noeud de secours.

Dans les deux cas, le programme d'exit est chargé d'initialiser et de démarrer tous les programmes qui étaient en cours d'exécution sur le noeud principal précédent, ce qui convertit le premier noeud de secours en nouveau noeud principal.

Par exemple, avec IBM MQ, le programme d'exit doit être en charge du démarrage du sous-système IBM MQ (QMQM) et des gestionnaires de files d'attente. Les gestionnaires de files d'attente doivent être configurés pour démarrer automatiquement les programmes d'écoute et les services, tels que les moniteurs de déclenchement.

Un exemple de programme d'exit, AMQSCRG4, est disponible sur IBM i.

## Configuration d'ASP indépendant commutable

IBM MQ peut être configuré pour tirer parti des capacités de mise en cluster de IBM i. Pour cela, procédez comme suit :

1. Création d'un cluster IBM i entre les systèmes de centre de données
2. Déplacez le gestionnaire de files d'attente vers un ASP indépendant.

[«Déplacement ou suppression d'un gestionnaire de files d'attente vers ou depuis un pool de mémoire secondaire indépendant»](#), à la page 438 contient un exemple de code pour vous aider à effectuer cette opération.

3. Vous devez créer un groupe de ressources en grappe définissant le domaine de reprise, l'ASP indépendant et le programme d'exit.

[«Configuration d'un groupe de ressources en grappe d'unité»](#), à la page 438 contient un exemple de code pour vous aider à effectuer cette opération.

### Concepts associés

[«ASP indépendants et haute disponibilité»](#), à la page 459

Les ASP indépendants permettent de déplacer des applications et des données entre les serveurs. La flexibilité des ASP indépendants signifie qu'ils constituent la base de certaines solutions à haute disponibilité IBM i . Lorsque vous envisagez d'utiliser un ASP ou un ASP indépendant pour le journal

du gestionnaire de files d'attente, vous devez prendre en compte une autre configuration à haute disponibilité basée sur des ASP indépendants.

### Configuration d'un groupe de ressources en grappe d'unité

Exemple de programme de configuration d'un groupe de ressources en grappe d'unité.

## Pourquoi et quand exécuter cette tâche

Dans l'exemple suivant, notez que:

- [ PRIMARY SITE NAME ] et [ BACKUP SITE NAME ] peuvent être deux chaînes distinctes de huit caractères ou moins.
- [ PRIMARY IP ] et [ BACKUP IP ] sont les adresses IP à utiliser pour la mise en miroir.

## Procédure

1. Identifiez le nom du cluster.
2. Identifiez le nom du programme d'exit du groupe de ressources en grappe et la bibliothèque.
3. Déterminez le nom du noeud principal et des noeuds de sauvegarde à définir par ce groupe de ressources en grappe.
4. Identifiez l'ASP indépendant à gérer par ce groupe de ressources en grappe et assurez-vous qu'il a été créé sous le noeud principal.
5. Créez une description d'unité dans les noeuds de sauvegarde à l'aide de la commande suivante:

```
CRTDEVASP DEVD([IASP NAME]) RSRNAME([IASP NAME])
```

6. Ajoutez l'adresse IP de relais à tous les noeuds à l'aide de la commande suivante:

```
ADDTCPIFC INTNETADR(' [TAKEOVER IP]') LIND([LINE DESC])  
SUBNETMASK(' [SUBNET MASK]') AUTOSTART(*NO)
```

7. Démarrez l'adresse IP de relais uniquement sur le noeud principal à l'aide de la commande suivante:

```
STRTCPIFC INTNETADR(' [TAKEOVER IP]')
```

8. Facultatif : Si votre ASP indépendant est commutable, appelez la commande suivante:

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT  
NAME])  
USRPRF([EXIT PROFILE]) RCDYMN(([PRIMARY NODE] *PRIMARY) ([BACKUP NAME] *BACKUP))  
EXITPGMFMFMT(EXTP0200) CFGOBJ([IASP NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

9. Facultatif : Si votre ASP indépendant doit être protégé par disque miroir, appelez la commande suivante:

```
CRTCRG CLUSTER([CLUSTER NAME]) CRG([CRG NAME]) CRGTYPE(*DEV) EXITPGM([EXIT LIB]/[EXIT NAME])  
USRPRF([EXIT PROFILE]) RCDYMN(([PRIMARY NODE] *PRIMARY *LAST [PRIMARY SITE NAME] (' [PRIMARY  
IP] '))  
[BACKUP NAME] *BACKUP *LAST [BACKUP SITE NAME] (' [BACKUP IP] '))) EXITPGMFMFMT(EXTP0200)  
CFGOBJ([IASP NAME] *DEVD *ONLINE '[TAKEOVER IP]')
```

### Déplacement ou suppression d'un gestionnaire de files d'attente vers ou depuis un pool de mémoire secondaire indépendant

Exemple de programme permettant de déplacer un gestionnaire de files d'attente vers un pool de mémoire secondaire indépendant (IASP) et de supprimer un gestionnaire de files d'attente d'un IASP.

## Pourquoi et quand exécuter cette tâche

Dans l'exemple suivant, notez que:

- [ MANAGER NAME ] est le nom de votre gestionnaire de files d'attente.
- [ IASP NAME ] est le nom de votre IASP.
- [ MANAGER LIBRARY ] est le nom de la bibliothèque de votre gestionnaire de files d'attente.
- [ MANAGER DIRECTORY ] est le nom de votre répertoire de gestionnaire de files d'attente.

## Procédure

1. Identifiez votre noeud principal et vos noeuds de sauvegarde.
2. Effectuez la procédure suivante sur votre noeud principal:
  - a) Vérifiez que votre gestionnaire de files d'attente est arrêté.
  - b) Vérifiez que votre ASP indépendant est vary on à l'aide de la commande

```
VRYCFG CFGOBJ([IASP NAME]) CFGTYPE(*DEV) STATUS(*ON)
```

- c) Créez le répertoire des gestionnaires de files d'attente sous l'ASP indépendant.  
Il y aura un répertoire sous la racine avec le nom de votre ASP indépendant, à savoir:

```
QSH CMD('mkdir -p /[IASP_NAME]/QIBM/UserData/mqm/qmgrs/')
```

- d) Déplacez les objets IFS de votre gestionnaire vers le répertoire des gestionnaires de files d'attente que vous venez de créer sous l'ASP indépendant à l'aide de la commande suivante:

```
QSH CMD('mv /QIBM/UserData/mqm/qmgrs/[MANAGER NAME]
/[IASP NAME]/QIBM/UserData/mqm/qmgrs')
```

- e) Créez un fichier sauvegarde temporaire nommé MGRLIB à l'aide de la commande suivante:

```
CRTSAVF QGPL/MGRLIB
```

- f) Sauvegardez votre bibliothèque de gestionnaires de files d'attente dans le fichier sauvegarde MGRLIB à l'aide de la commande suivante:

```
SAVLIB LIB([MANGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
```

- g) Supprimez la bibliothèque du gestionnaire de files d'attente à l'aide de la commande suivante et ignorez tous les messages d'interrogation:

```
DLTLIB [MANAGER LIBRARY]
```

- h) Restaurez la bibliothèque de votre gestionnaire de files d'attente dans l'ASP indépendant à l'aide de la commande suivante:

```
RSTLIB SAVLIB([MANAGER LIBRARY]) DEV(*SAVF) SAVF(QGPL/MGRLIB)
RSTASPDEV([IASP NAME])
```

- i) Supprimez le fichier sauvegarde temporaire à l'aide de la commande suivante:

```
DLTF FILE(QGPL/MGRLIB)
```

- j) Créez un lien symbolique vers les objets IFS du gestionnaire de files d'attente sous l'ASP indépendant à l'aide de la commande suivante:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

- k) Effectuez une connexion à l'ASP indépendant à l'aide de la commande suivante:

```
SETASPGRP [IASP NAME]
```

- l) Démarrez votre gestionnaire de files d'attente à l'aide de la commande suivante:

```
STRMQM [MANAGER NAME]
```

3. Exécutez la procédure suivante sur votre ou vos noeuds de sauvegarde:

a) Créez un répertoire de gestionnaire de files d'attente temporaire à l'aide de la commande suivante:

```
QSH CMD('mkdir -p /[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

b) Créez un lien symbolique vers le répertoire temporaire du gestionnaire de files d'attente à l'aide de la commande suivante:

```
ADDLNK OBJ('/[IASP NAME]/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')  
NEWLNK('/QIBM/UserData/mqm/qmgrs/[MANAGER NAME]')
```

c) Supprimez le répertoire temporaire à l'aide de la commande suivante:

```
QSH CMD('rm -r /[IASP NAME]')
```

d) Ajoutez ce qui suit à la fin du fichier /QIBM/UserData/mqm/mqs.ini:

```
QueueManager:  
Name=[MANAGER NAME]  
Prefix=/QIBM/UserData/mqm  
Library=[MANAGER LIBRARY]  
Directory=[MANAGER DIRECTORY]
```

4. Pour supprimer un gestionnaire de files d'attente d'un ASP indépendant, exécutez les commandes suivantes:

a) VRYCFG CFGOBJ ([ NOM FERMOIR ]) CFGTYPE (\*DEV) STATUS (\*ON)

b) SETASPGRP [ NOM ASP indépendant ]

c) ENDMQM [ NOM DU GERE ]

d) DLTMQM [ NOM DU GERE ]

## **Configuration de journal protégé par disque miroir pour ASP sous IBM i**

Configurez un gestionnaire de files d'attente multi-instance robuste à l'aide de la réplication synchrone entre les journaux en miroir.

Une configuration de gestionnaire de files d'attente protégée par disque miroir utilise des journaux créés dans des pools de mémoire secondaire (ASP) de base ou indépendants.

Sous IBM i, les données du gestionnaire de files d'attente sont écrites dans des journaux et dans un système de fichiers. Les journaux contiennent la copie principale des données du gestionnaire de files d'attente. Les journaux sont partagés entre les systèmes à l'aide de la réplication de journal synchrone ou asynchrone. Une combinaison de journaux locaux et éloignés est requise pour redémarrer une instance de gestionnaire de files d'attente. Le redémarrage du gestionnaire de files d'attente lit les enregistrements de journal à partir de la combinaison de journaux locaux et éloignés sur le serveur et des données du gestionnaire de files d'attente sur le système de fichiers réseau partagé. Les données du système de fichiers accélèrent le redémarrage du gestionnaire de files d'attente. Les points de contrôle sont stockés dans le système de fichiers, marquant les points de synchronisation entre le système de fichiers et les journaux. Les enregistrements de journal stockés avant le point de contrôle ne sont pas requis pour les redémarrages classiques du gestionnaire de files d'attente. Toutefois, il se peut que les données du système de fichiers ne soient pas à jour et que les enregistrements de journal après le point de contrôle soient utilisés pour terminer le redémarrage du gestionnaire de files d'attente. Les données des journaux associés à l'instance sont maintenues à jour afin que le redémarrage aboutisse.

Mais même les enregistrements de journal peuvent ne pas être à jour, si le journal distant sur le serveur de secours était répliqué de manière asynchrone et que l'incident s'est produit avant qu'il ne soit synchronisé. Si vous décidez de redémarrer un gestionnaire de files d'attente à l'aide d'un journal éloigné qui n'est pas synchronisé, l'instance de gestionnaire de files d'attente de secours peut soit retraiter les



messages qui ont été supprimés avant l'échec de l'instance active, soit ne pas traiter les messages qui ont été reçus avant l'échec de l'instance active.

Une autre possibilité, rare, est que le système de fichiers contienne l'enregistrement de point de contrôle le plus récent et qu'un journal éloigné non synchronisé sur la base de données de secours ne le contienne pas. Dans ce cas, le gestionnaire de files d'attente ne redémarre pas automatiquement. Vous avez le choix entre attendre que le journal éloigné soit synchronisé ou démarrer à froid le gestionnaire de files d'attente de secours à partir du système de fichiers. Même si, dans ce cas, le système de fichiers contient un point de contrôle plus récent des données du gestionnaire de files d'attente que le journal éloigné, il peut ne pas contenir tous les messages qui ont été traités avant l'échec de l'instance active. Certains messages peuvent être traités à nouveau et d'autres non traités après un redémarrage à froid qui n'est pas synchronisé avec les journaux.

Avec un gestionnaire de files d'attente multi-instance, le système de fichiers est également utilisé pour contrôler l'instance d'un gestionnaire de files d'attente qui est active et celle de secours. L'instance active acquiert un verrou sur les données du gestionnaire de files d'attente. La base de données de secours attend pour acquérir le verrou, et lorsqu'elle le fait, elle devient l'instance active. Le verrou est libéré par l'instance active, s'il se termine normalement. Le verrou est libéré par le système de fichiers si le système de fichiers détecte que l'instance active a échoué ou qu'il ne peut pas accéder au système de fichiers. Le système de fichiers doit répondre aux conditions requises pour la détection des incidents ; voir [Conditions requises pour les systèmes de fichiers partagés](#).

L'architecture des gestionnaires de files d'attente multi-instance sous IBM i permet le redémarrage automatique après une défaillance du serveur ou du gestionnaire de files d'attente. Il prend également en charge la restauration des données du gestionnaire de files d'attente suite à une défaillance du système de fichiers dans lequel les données du gestionnaire de files d'attente sont stockées.

Dans [Figure 24](#), à la [page 442](#), si ALPHA échoue, vous pouvez redémarrer manuellement QM1 sur BETA, à l'aide du journal en miroir. En ajoutant la fonction de gestionnaire de files d'attente multi-instance à QM1, l'instance de secours de QM1 reprend automatiquement sur BETA si l'instance active sur ALPHA échoue. QM1 peut également reprendre automatiquement si c'est le serveur ALPHA qui échoue, et pas seulement l'instance active de QM1. Une fois que BETA devient l'hôte de l'instance de gestionnaire de files d'attente active, l'instance de secours peut être démarrée sur ALPHA.

La [Figure 24](#), à la [page 442](#) présente une configuration qui met en miroir les journaux entre deux instances d'un gestionnaire de files d'attente à l'aide de NetServer pour stocker les données du gestionnaire de files d'attente. Vous pouvez développer le modèle pour inclure plus de journaux, et donc plus d'instances. Suivez les règles de dénomination de journal décrites dans la rubrique «[Journaux de gestionnaire de files d'attente sous IBM i](#)», à la [page 419](#). Actuellement, le nombre d'instances en cours d'exécution d'un gestionnaire de files d'attente est limité à deux, une est active et une est en veille.

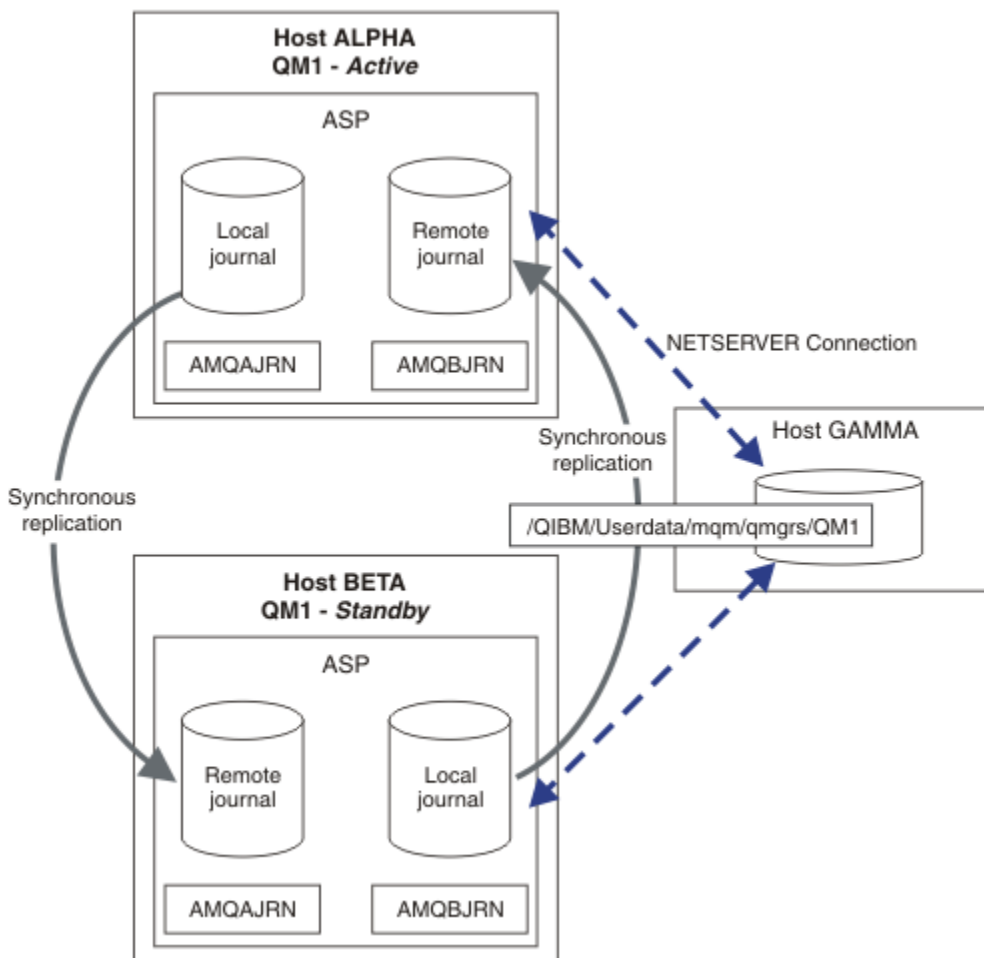


Figure 24. Mettre en miroir un journal de gestionnaire de files d'attente

Le journal local pour QM1 sur l'hôte ALPHA est appelé AMQAJRN (ou plus complètement, QMQM1/AMQAJRN) et sur BETA le journal est QMQM1/AMQBJRN. Chaque journal local est répliqué dans des journaux éloignés sur toutes les autres instances du gestionnaire de files d'attente. Si le gestionnaire de files d'attente est configuré avec deux instances, un journal local est répliqué sur un journal distant.

### \*SYNC ou \*ASYNCRéplication de journal éloigné

Les journaux IBM i sont mis en miroir à l'aide de l'une des deux options suivantes: synchrone (\*SYNC) ou asynchrone (\*ASYNCR) journalisation; voir [Remote journal management](#).

Le mode de réplication dans [Figure 24](#), à la page 442 est \*SYNC, et non \*ASYNCR. \*ASYNCR est plus rapide, mais si un incident se produit lorsque l'état du journal éloigné est \*ASYNCRPEND, le journal local et le journal éloigné ne sont pas cohérents. Le journal éloigné doit rattraper le journal local. Si vous choisissez \*SYNC, le système local attend le journal éloigné avant de renvoyer un appel nécessitant une écriture terminée. Les journaux locaux et éloignés restent généralement cohérents les uns avec les autres. Uniquement si l'opération \*SYNC prend plus de temps qu'une heure définie<sup>1</sup>, et la journalisation à distance est désactivée, les journaux sont désynchronisés. Une erreur est consignée dans la file d'attente de messages du journal et dans QSYSOPR. Le gestionnaire de files d'attente détecte ce message, écrit une erreur dans le journal des erreurs du gestionnaire de files d'attente et désactive la réplication à distance du journal du gestionnaire de files d'attente. L'instance de gestionnaire de files d'attente active reprend sans journalisation à distance dans ce journal. Lorsque le serveur distant est à nouveau

<sup>1</sup> La durée indiquée est de 60 secondes sur IBM i 5 et est comprise entre 1 et 3600 secondes sur IBM i 6.1 et les versions suivantes.

disponible, vous devez réactiver manuellement la réplication de journal distant synchrone. Les journaux sont ensuite resynchronisés.

Un problème lié à la configuration \*SYNC / \*SYNC illustrée dans la Figure 24, à la page 442 est la façon dont l'instance de gestionnaire de files d'attente de secours sur BETA prend le contrôle. Dès que l'instance de gestionnaire de files d'attente sur BETA écrit son premier message persistant, elle tente de mettre à jour le journal éloigné sur ALPHA. Si la cause de la transmission du contrôle de ALPHA à BETA est l'échec de ALPHA et que ALPHA est toujours arrêté, la journalisation à distance dans ALPHA échoue. BETA attend la réponse de ALPHA, puis désactive la journalisation à distance et reprend le traitement des messages avec uniquement la journalisation locale. BETA doit attendre un certain temps pour détecter que ALPHA est arrêté, provoquant une période d'inactivité.

Le choix entre la définition de la journalisation à distance sur \*SYNC ou \*ASYNCR est une option de compensation. Le Tableau 24, à la page 443 récapitule les compromis entre l'utilisation de la journalisation \*SYNC et \*ASYNCR entre une paire de gestionnaires de files d'attente:

Actif	Secours	*SYNC	*ASYNCR
*SYNC		<ol style="list-style-type: none"> <li>1. Commutation et reprise en ligne cohérentes</li> <li>2. L'instance de secours ne reprend pas immédiatement après la reprise en ligne.</li> <li>3. La journalisation à distance doit être disponible en permanence</li> <li>4. Les performances du gestionnaire de files d'attente dépendent de la journalisation à distance</li> </ol>	<ol style="list-style-type: none"> <li>1. Commutation et reprise en ligne cohérentes</li> <li>2. La journalisation à distance doit être commutée sur *SYNC lorsque le serveur de secours est disponible</li> <li>3. La journalisation à distance doit rester disponible une fois qu'elle a été redémarrée</li> <li>4. Les performances du gestionnaire de files d'attente dépendent de la journalisation à distance</li> </ol>
*ASYNCR		<ol style="list-style-type: none"> <li>1. Pas une combinaison sensée</li> </ol>	<ol style="list-style-type: none"> <li>1. Certains messages peuvent être perdus ou dupliqués après une reprise en ligne ou une commutation</li> <li>2. Il n'est pas nécessaire que l'instance de secours soit disponible en permanence pour que l'instance active puisse continuer sans délai.</li> <li>3. Les performances ne dépendent pas de la journalisation à distance</li> </ol>

#### \*SYNC / \*SYNC

L'instance de gestionnaire de files d'attente active utilise la journalisation \*SYNC et, lorsque l'instance de gestionnaire de files d'attente de secours démarre, elle tente immédiatement d'utiliser la journalisation \*SYNC .

1. Le journal éloigné est cohérent au niveau transactionnel avec le journal local du gestionnaire de files d'attente actif. Si le gestionnaire de files d'attente est basculé sur l'instance de secours, il peut reprendre immédiatement. L'instance de secours reprend normalement sans perte ni duplication des messages. Les messages ne sont perdus ou dupliqués que si la journalisation à distance a échoué depuis le dernier point de contrôle et que le gestionnaire de files d'attente précédemment actif ne peut pas être redémarré.
2. Si le gestionnaire de files d'attente bascule sur l'instance de secours, il se peut qu'il ne puisse pas démarrer immédiatement. L'instance de gestionnaire de files d'attente de secours est activée avec la journalisation \*SYNC . La cause de la reprise en ligne peut empêcher la journalisation à distance sur le serveur hébergeant l'instance de secours. Le gestionnaire de files d'attente

attend que le problème soit détecté avant de traiter les messages persistants. Une erreur est consignée dans la file d'attente de messages du journal et dans QSYSOPR. Le gestionnaire de files d'attente détecte ce message, écrit une erreur dans le journal des erreurs du gestionnaire de files d'attente et désactive la réplication à distance du journal du gestionnaire de files d'attente. L'instance de gestionnaire de files d'attente active reprend sans journalisation à distance dans ce journal. Lorsque le serveur distant est à nouveau disponible, vous devez réactiver manuellement la réplication de journal distant synchrone. Les journaux sont ensuite resynchronisés.

3. Le serveur sur lequel le journal distant est répliqué doit toujours être disponible pour gérer le journal distant. Le journal distant est généralement répliqué sur le même serveur que celui qui héberge le gestionnaire de files d'attente de secours. Il se peut que le serveur ne soit plus disponible. Une erreur est consignée dans la file d'attente de messages du journal et dans QSYSOPR. Le gestionnaire de files d'attente détecte ce message, écrit une erreur dans le journal des erreurs du gestionnaire de files d'attente et désactive la réplication à distance du journal du gestionnaire de files d'attente. L'instance de gestionnaire de files d'attente active reprend sans journalisation à distance dans ce journal. Lorsque le serveur distant est à nouveau disponible, vous devez réactiver manuellement la réplication de journal distant synchrone. Les journaux sont ensuite resynchronisés.
4. La journalisation à distance est plus lente que la journalisation locale, et beaucoup plus lente si les serveurs sont séparés par une grande distance. Le gestionnaire de files d'attente doit attendre la journalisation à distance, ce qui réduit les performances du gestionnaire de files d'attente.

La configuration \*SYNC / \*SYNC entre deux serveurs présente l'inconvénient de retarder la reprise de l'instance de secours après la reprise en ligne. La configuration \*SYNC / \*ASYNCR ne présente pas cet incident.

\*SYNC / \*ASYNCR ne garantit aucune perte de message après une commutation ou une reprise en ligne, tant qu'un journal éloigné est disponible. Si vous souhaitez réduire le risque de perte de message après une reprise en ligne ou une commutation, vous disposez de deux options. Arrêtez l'instance active si le journal éloigné devient inactif ou créez des journaux éloignés sur plusieurs serveurs.

#### **\*SYNC / \*ASYNCR**

L'instance de gestionnaire de files d'attente active utilise la journalisation \*SYNC et, lorsque l'instance de gestionnaire de files d'attente de secours démarre, elle utilise la journalisation \*ASYNCR. Peu de temps après la disponibilité du serveur hébergeant la nouvelle instance de secours, l'opérateur système doit basculer le journal éloigné sur l'instance active vers \*SYNC. Lorsque l'opérateur fait passer la journalisation éloignée de \*ASYNCR à \*SYNC, l'instance active s'interrompt si l'état du journal éloigné est \*ASYNCRPEND. L'instance de gestionnaire de files d'attente active attend que les postes de journal restants soient transférés vers le journal éloigné. Lorsque le journal éloigné a été synchronisé avec le journal local, la nouvelle base de données de secours est à nouveau cohérente au niveau transactionnel avec la nouvelle instance active. Du point de vue de la gestion des gestionnaires de files d'attente multi-instance, dans une configuration \*SYNC / \*ASYNCR, l'opérateur système IBM i a une tâche supplémentaire. L'opérateur doit basculer la journalisation à distance vers \*SYNC en plus du redémarrage de l'instance de gestionnaire de files d'attente défaillante.

1. Le journal éloigné est cohérent au niveau transactionnel avec le journal local du gestionnaire de files d'attente actif. Si l'instance de gestionnaire de files d'attente active est basculée ou bascule sur l'instance de secours, l'instance de secours peut reprendre immédiatement. L'instance de secours reprend normalement sans perte ni duplication des messages. Les messages ne sont perdus ou dupliqués que si la journalisation à distance a échoué depuis le dernier point de contrôle et que le gestionnaire de files d'attente précédemment actif ne peut pas être redémarré.
2. L'opérateur système doit basculer le journal éloigné de \*ASYNCR vers \*SYNC peu après que le système hébergeant l'instance active soit à nouveau disponible. L'opérateur peut attendre le rattrapage du journal éloigné avant de passer à \*SYNC. L'opérateur peut également basculer immédiatement l'instance distante sur \*SYNC et forcer l'instance active à attendre que le journal de l'instance de secours soit rattrapé. Lorsque la journalisation à distance est définie sur \*SYNC, l'instance de secours est généralement cohérente au niveau transactionnel avec l'instance active. Les messages ne sont perdus ou dupliqués que si la journalisation à distance a échoué depuis le dernier point de contrôle et que le gestionnaire de files d'attente précédemment actif ne peut pas être redémarré.

3. Lorsque la configuration a été restaurée à partir d'une commutation ou d'une reprise en ligne, le serveur sur lequel le journal éloigné est hébergé doit être disponible en permanence.

Choisissez \*SYNC / \*ASYNCR lorsque vous souhaitez que le gestionnaire de files d'attente de secours reprenne rapidement après une reprise en ligne. Vous devez restaurer manuellement le paramètre de journal éloigné sur \*SYNC sur la nouvelle instance active. La configuration \*SYNC / \*ASYNCR correspond au modèle normal d'administration d'une paire de gestionnaires de files d'attente multi-instance. Une fois qu'une instance a échoué, il y a un délai avant le redémarrage de l'instance de secours, au cours duquel l'instance active ne peut pas basculer.

#### **\*ASYNCR / \*ASYNCR**

Les deux serveurs hébergeant les gestionnaires de files d'attente actifs et de secours sont configurés pour utiliser la journalisation à distance \*ASYNCR .

1. Lors de la commutation ou de la reprise en ligne, le gestionnaire de files d'attente continue avec le journal sur le nouveau serveur. Il se peut que le journal ne soit pas synchronisé lors de la commutation ou de la reprise en ligne. Par conséquent, des messages peuvent être perdus ou dupliqués.
2. L'instance active s'exécute, même si le serveur hébergeant le gestionnaire de files d'attente de secours n'est pas disponible. Le journal local est répliqué de manière asynchrone avec le serveur de secours lorsqu'il est disponible.
3. Les performances du gestionnaire de files d'attente local ne sont pas affectées par la journalisation à distance.

Choisissez \*ASYNCR / \*ASYNCR si les performances sont votre exigence principale et que vous êtes prêt à perdre ou à dupliquer certains messages après la reprise en ligne ou la commutation.

#### **\*ASYNCR / \*SYNC**

Il n'y a aucune raison d'utiliser cette combinaison d'options.

### **Activation du gestionnaire de files d'attente à partir d'un journal éloigné**

Les journaux sont répliqués de manière synchrone ou asynchrone. Il se peut que le journal éloigné ne soit pas actif ou qu'il rattrape le journal local. Le journal distant peut être en cours de rattrapage, même s'il est répliqué de manière synchrone, car il a peut-être été récemment activé. Les règles que le gestionnaire de files d'attente applique à l'état du journal éloigné qu'il utilise lors du démarrage sont les suivantes.

1. Le démarrage de la base de données de secours échoue si elle doit être relecture à partir du journal éloigné sur la base de données de secours et que le statut du journal est \*FAILED ou \*INACTPEND.
2. Lorsque l'activation de la base de données de secours commence, l'état du journal éloigné sur la base de données de secours doit être \*ACTIVE ou \*INACTIVE. Si l'état est \*INACTIVE, l'activation peut échouer si toutes les données de journal n'ont pas été répliquées.

L'échec se produit si les données du gestionnaire de files d'attente sur le système de fichiers réseau contiennent un enregistrement de point de contrôle plus récent que celui présent dans le journal éloigné. Il est peu probable que l'échec se produise tant que le journal éloigné est bien activé dans l'intervalle maximal de 30 minutes par défaut entre les points de contrôle. Si le gestionnaire de files d'attente de secours lit un enregistrement de point de contrôle plus récent à partir du système de fichiers, il ne démarre pas.

Vous avez le choix: attendez que le journal local sur le serveur actif puisse être restauré ou démarrez à froid le gestionnaire de files d'attente de secours. Si vous choisissez le démarrage à froid, le gestionnaire de files d'attente démarre sans données de journal et repose sur la cohérence et l'exhaustivité des données du gestionnaire de files d'attente dans le système de fichiers.

**Remarque :** Si vous démarrez un gestionnaire de files d'attente à froid, vous risquez de perdre ou de dupliquer des messages après le dernier point de contrôle. Les transactions de message ont été écrites dans le journal, mais certaines des transactions n'ont peut-être pas été écrites dans les données du gestionnaire de files d'attente dans le système de fichiers. Lorsque vous démarrez un gestionnaire de files d'attente à froid, un nouveau journal est démarré et les transactions qui ne sont pas écrites dans les données du gestionnaire de files d'attente du système de fichiers sont perdues.

3. L'activation du gestionnaire de files d'attente de secours attend que l'état du journal éloigné sur le serveur de secours passe de \*ASYNCPEND ou \*SYNCPEND à \*ASYNCR ou \*SYNCR. Des messages sont écrits périodiquement dans le journal des travaux du contrôleur d'exécution.

**Remarque :** Dans ce cas, l'activation est en attente sur le journal éloigné local du gestionnaire de files d'attente de secours en cours d'activation. Le gestionnaire de files d'attente attend également un certain temps avant de continuer sans journal éloigné. Il attend lorsqu'il tente d'écrire de manière synchrone dans son ou ses journaux éloignés et que le journal n'est pas disponible.

4. L'activation s'arrête si le journal passe à l'état \*FAILED ou \*INACTPEND.

Les noms et les états des journaux locaux et éloignés à utiliser dans l'activation sont consignés dans le journal des erreurs du gestionnaire de files d'attente.

**IBM i** *Création d'un gestionnaire de files d'attente multi-instance à l'aide de la mise en miroir du journal et de NetServer sous IBM i*

Créez un gestionnaire de files d'attente multi-instance à exécuter sur deux serveurs IBM i . Les données du gestionnaire de files d'attente sont stockées sur un troisième serveur IBM i à l'aide de NetServer. Le journal du gestionnaire de files d'attente est mis en miroir entre les deux serveurs à l'aide de la journalisation éloignée. La commande **ADDQMJRN** permet de simplifier la création des journaux éloignés.

## Avant de commencer

1. La tâche requiert trois serveurs IBM i . Installez IBM MQ sur deux d'entre eux, ALPHA et BETA dans l'exemple. le produit doit être au moins au niveau IBM WebSphere MQ 7.0.1 Fix Pack 1.
2. Le troisième serveur est un serveur IBM i , connecté par NetServer à ALPHA et BETA. Il est utilisé pour partager les données du gestionnaire de files d'attente. Il n'est pas nécessaire d'avoir une installation IBM MQ . Il est utile d'installer IBM MQ sur le serveur en tant qu'étape temporaire pour configurer les répertoires et les droits du gestionnaire de files d'attente.
3. Assurez-vous que le profil utilisateur QMQM possède le même mot de passe sur les trois serveurs.
4. Installez IBM i NetServer; voir [i5/OS NetServer](#).

## Pourquoi et quand exécuter cette tâche

Procédez comme suit pour créer la configuration illustrée dans la [Figure 25](#), à la [page 449](#). Les données du gestionnaire de files d'attente sont connectées à l'aide d' IBM i NetServer.

- Créez des connexions depuis ALPHA et BETA vers le partage de répertoire sur GAMMA qui permet de stocker les données du gestionnaire de files d'attente. La tâche configure également les droits d'accès, les profils utilisateur et les mots de passe nécessaires.
- Ajoutez des entrées de base de données relationnelle (RDBE) aux systèmes IBM i qui vont exécuter des instances de gestionnaire de files d'attente. Les entrées RDBE sont utilisées pour la connexion aux systèmes IBM i utilisés pour la journalisation à distance.
- Créez le gestionnaire de files d'attente QM1 sur le serveur IBM i , ALPHA.
- Ajoutez les informations de contrôle du gestionnaire de files d'attente pour QM1 sur l'autre serveur IBM i , BETA.
- Créez des journaux éloignés sur les deux serveurs IBM i pour les deux instances de gestionnaire de files d'attente. Chaque gestionnaire de files d'attente écrit dans le journal local. Le journal local est répliqué dans le journal éloigné. La commande **ADDQMJRN** simplifie l'ajout des journaux et des connexions.
- Démarrez le gestionnaire de files d'attente en autorisant une instance de secours.

## Procédure

1. Exécutez la tâche «[Création d'un partage de réseau pour les données de gestionnaire de files d'attente à l'aide de NetServer sous IBM i](#)», à la [page 432](#).

Par conséquent, ALPHA et BETA ont un partage, /QNTC/GAMMA/WMQ, qui pointe vers /QIBM/UserData/mqm/qmgrs sur GAMMA. Les profils utilisateur QMQM et QMQMADM disposent des droits nécessaires et QMQM possède des mots de passe correspondants sur les trois systèmes.

2. Ajoutez des entrées de base de données relationnelle (RDBE) aux systèmes IBM i qui vont héberger des instances de gestionnaire de files d'attente.

a) Sur ALPHA, créez la connexion à BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

b) Sur BETA, créez les connexions à ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Créez le gestionnaire de files d'attente QM1 sur ALPHA, en sauvegardant les données du gestionnaire de files d'attente sur GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ')
```

Le chemin d'accès utilise NetServer pour créer les données du gestionnaire de files d'attente.

4. Exécuter sur ALPHA. La commande ajoute un journal éloigné sur BETA pour.

```
ADDMQMJRN MQMNAME(QM1) RMTJRNRDB(BETA)
```

crée des entrées de journal dans son journal local sur ALPHA lorsque l'instance active est sur ALPHA. Le journal local sur ALPHA est répliqué sur le journal éloigné sur BETA.

5. Utilisez la commande, pour inspecter les données de configuration IBM MQ créées par pour sur ALPHA.

Les informations sont nécessaires à l'étape suivante.

Dans cet exemple, la configuration suivante est créée dans ALPHA pour:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

6. Créez une instance de gestionnaire de files d'attente de QM1 sur BETA à l'aide de la commande. Exécutez la commande suivante sur BETA pour modifier les informations de contrôle du gestionnaire de files d'attente sur BETA.

```
ADDMQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH(' /QNTC/GAMMA/WMQ /QM1')
```

**Conseil :** Copiez et collez les informations de configuration. La section du gestionnaire de files d'attente est la même sur ALPHA et BETA.

7. Exécuter sur BETA. La commande ajoute un journal local sur BETA et un journal éloigné sur ALPHA pour.

```
ADDMQMJRN MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

crée des entrées de journal dans son journal local sur BETA lorsque l'instance active est sur BETA. Le journal local sur BETA est répliqué sur le journal éloigné sur ALPHA.

**Remarque :** Vous pouvez également configurer la journalisation à distance de BETA à ALPHA à l'aide de la journalisation asynchrone.

Utilisez cette commande pour configurer la journalisation asynchrone de BETA vers ALPHA, à la place de la commande de l'étape «7», à la page 447.

```
ADDMQMJRN MQMNAME (QM1) RMTJRNRDB (ALPHA) RMTJRNDLV (*ASYNC)
```

Si le serveur ou la journalisation sur ALPHA est la source de l'échec, la fonction BETA démarre sans attendre que de nouvelles entrées de journal soient répliquées sur ALPHA.

Basculez le mode de réplication sur \*SYNC, à l'aide de la commande, lorsque ALPHA est à nouveau en ligne.

Utilisez les informations de la rubrique «[Configuration de journal protégé par disque miroir pour ASP sous IBM i](#)», à la page 440 pour déterminer si les journaux doivent être mis en miroir de manière synchrone, asynchrone ou mixte. Par défaut, la réplication est synchrone, avec une période d'attente de 60 secondes pour une réponse du journal éloigné.

8. Vérifiez que les journaux ALPHA et BETA sont activés et que le statut de la réplication de journal éloigné est activé.

a) Sur ALPHA:

```
WRKMQMJRN MQMNAME (QM1)
```

b) Sur BETA:

```
WRKMQMJRN MQMNAME (QM1)
```

9. Démarrez les instances de gestionnaire de files d'attente sur ALPHA et BETA.

a) Démarrez la première instance sur ALPHA, ce qui en fait l'instance active. Activation du basculement vers une instance de secours.

```
STRMQM MQMNAME (QM1) STANDBY (*YES)
```

b) Démarrez la deuxième instance sur BETA, ce qui en fait l'instance de secours.

```
STRMQM MQMNAME (QM1) STANDBY (*YES)
```

## Résultats

Permet de vérifier le statut du gestionnaire de files d'attente:

1. Le statut de l'instance de gestionnaire de files d'attente sur ALPHA doit être.
2. Le statut de l'instance de gestionnaire de files d'attente sur BETA doit être.



## Exemple

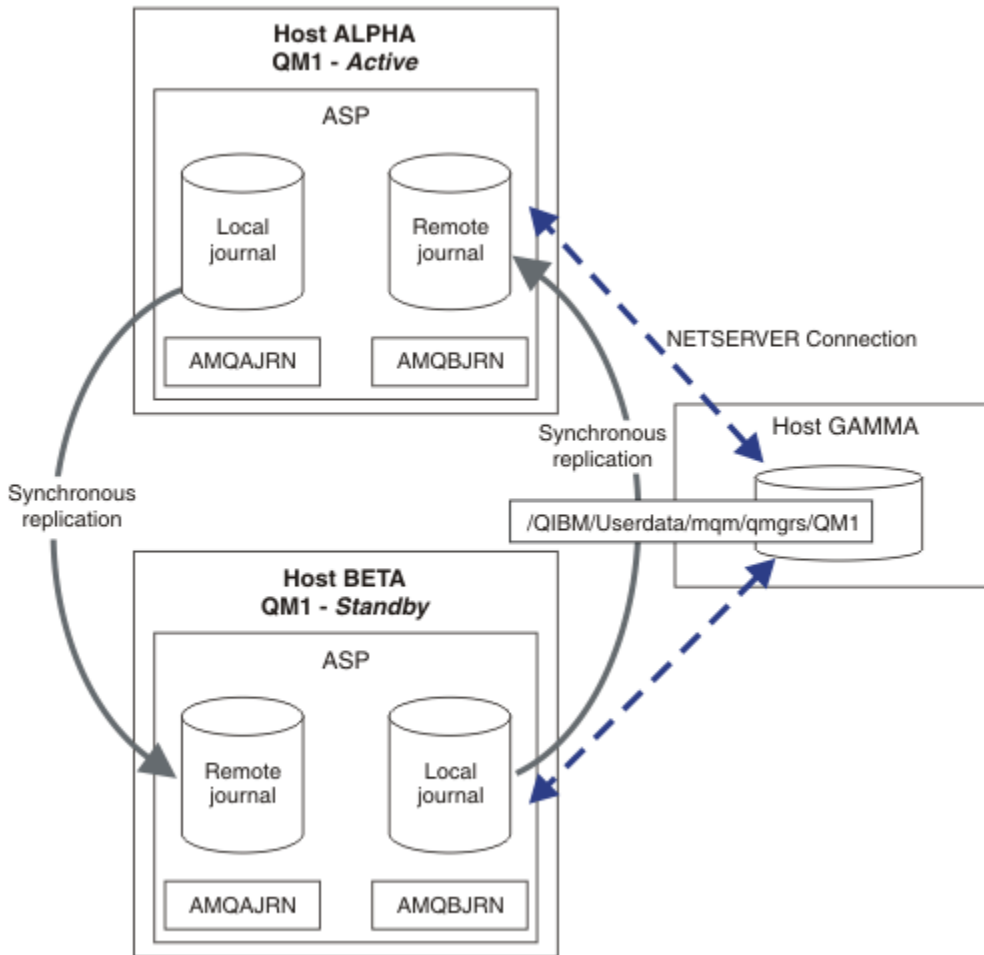


Figure 25. Configuration du journal en miroir

## Que faire ensuite

- Vérifiez que les instances actives et de secours basculent automatiquement. Vous pouvez exécuter les exemples de programmes à haute disponibilité pour tester le basculement ; voir [Exemples de programmes à haute disponibilité](#). Les exemples de programme sont des clients 'C'. Vous pouvez les exécuter à partir d'une plateforme Windows ou Unix.

1. Démarrez les exemples de programmes à haute disponibilité.
2. Sous ALPHA, arrêtez le gestionnaire de files d'attente demandant le basculement:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Vérifiez que l'instance de sur BETA est active.
4. Redémarrer sur ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Examinez les autres configurations à haute disponibilité:
  1. Utilisez NetServer pour placer les données du gestionnaire de files d'attente sur un serveur Windows .

2. Au lieu d'utiliser la journalisation éloignée pour reproduire le journal du gestionnaire de files d'attente, stockez le journal sur un ASP indépendant. Utilisez la mise en grappe IBM i pour transférer l'ASP indépendant de ALPHA vers BETA.

### **IBM i** Conversion d'un gestionnaire de files d'attente à instance unique en gestionnaire de files d'attente multi-instance à l'aide de NetServer et de la mise en miroir du journal sous IBM i

Convertissez un gestionnaire de files d'attente à instance unique en gestionnaire de files d'attente multi-instance. Déplacez les données du gestionnaire de files d'attente vers un partage de réseau connecté par NetServer. Reproduisez le journal du gestionnaire de files d'attente sur un second serveur IBM i à l'aide de la journalisation à distance.

## Avant de commencer

1. La tâche requiert trois serveurs IBM i . L'installation IBM MQ existante, sur le serveur ALPHA dans l'exemple, doit être au moins au niveau de IBM WebSphere MQ 7.0.1 Fix Pack 1. ALPHA exécute un gestionnaire de files d'attente appelé QM1 dans l'exemple.
2. Installez IBM MQ sur le second serveur IBM i , BETA dans l'exemple.
3. Le troisième serveur est un serveur IBM i , connecté par NetServer à ALPHA et BETA. Il est utilisé pour partager les données du gestionnaire de files d'attente. Il n'est pas nécessaire d'avoir une installation IBM MQ . Il est utile d'installer IBM MQ sur le serveur en tant qu'étape temporaire pour configurer les répertoires et les droits du gestionnaire de files d'attente.
4. Assurez-vous que le profil utilisateur QMQM possède le même mot de passe sur les trois serveurs.
5. Installez IBM i NetServer; voir [i5/OS NetServer](#).

## Pourquoi et quand exécuter cette tâche

Procédez comme suit pour convertir un gestionnaire de files d'attente à instance unique en gestionnaire de files d'attente multi-instance indiqué dans la [Figure 26](#), à la [page 454](#). Le gestionnaire de files d'attente à instance unique est supprimé dans la tâche, puis recréé, stockant les données du gestionnaire de files d'attente sur le partage de réseau connecté par NetServer. Cette procédure est plus fiable que le déplacement des répertoires et des fichiers du gestionnaire de files d'attente vers le partage de réseau à l'aide de la commande **CPY** .

- Créez des connexions depuis ALPHA et BETA vers le partage de répertoire sur GAMMA qui permet de stocker les données du gestionnaire de files d'attente. La tâche configure également les droits d'accès, les profils utilisateur et les mots de passe nécessaires.
- Ajoutez des entrées de base de données relationnelle (RDBE) aux systèmes IBM i qui vont exécuter des instances de gestionnaire de files d'attente. Les entrées RDBE sont utilisées pour la connexion aux systèmes IBM i utilisés pour la journalisation à distance.
- Sauvegardez les journaux et les définitions du gestionnaire de files d'attente, arrêtez le gestionnaire de files d'attente et supprimez-le.
- Recréez le gestionnaire de files d'attente en stockant les données du gestionnaire de files d'attente sur le partage de réseau sur GAMMA.
- Ajoutez la deuxième instance du gestionnaire de files d'attente à l'autre serveur.
- Créez des journaux éloignés sur les deux serveurs IBM i pour les deux instances de gestionnaire de files d'attente. Chaque gestionnaire de files d'attente écrit dans le journal local. Le journal local est répliqué dans le journal éloigné. La commande **ADDQMJRN** simplifie l'ajout des journaux et des connexions.
- Démarrez le gestionnaire de files d'attente en autorisant une instance de secours.

### Remarque :

A l'étape «4», à la [page 451](#) de la tâche, vous supprimez le gestionnaire de files d'attente à instance unique, QM1. La suppression du gestionnaire de files d'attente supprime tous les messages persistants dans les files d'attente. Pour cette raison, terminez le traitement de tous les messages stockés par le gestionnaire de files d'attente, avant de convertir le gestionnaire de files d'attente. Si le traitement de tous les messages n'est pas possible, sauvegardez la bibliothèque du gestionnaire de files d'attente avant

l'étape «4», à la page 451. Restaurez la bibliothèque du gestionnaire de files d'attente après l'étape «5», à la page 451.

### Remarque :

A l'étape «5», à la page 451 de la tâche, vous recréez QM1. Bien que le gestionnaire de files d'attente ait le même nom, il possède un identificateur de gestionnaire de files d'attente différent. La mise en cluster du gestionnaire de files d'attente utilise l'identificateur de gestionnaire de files d'attente. Pour supprimer et recréer un gestionnaire de files d'attente dans un cluster, vous devez d'abord supprimer le gestionnaire de files d'attente du cluster ; voir [Suppression d'un gestionnaire de files d'attente d'un cluster: méthode alternative](#) ou [Suppression d'un gestionnaire de files d'attente d'un cluster](#). Une fois que vous avez recréé le gestionnaire de files d'attente, ajoutez-le au cluster. Bien qu'il porte le même nom qu'auparavant, il semble s'agir d'un nouveau gestionnaire de files d'attente pour les autres gestionnaires de files d'attente du cluster.

## Procédure

1. Exécutez la tâche «Création d'un partage de réseau pour les données de gestionnaire de files d'attente à l'aide de NetServer sous IBM i», à la page 432.

Par conséquent, ALPHA et BETA ont un partage, /QNTC/GAMMA/WMQ, qui pointe vers /QIBM/UserData/mqm/qmgrs sur GAMMA. Les profils utilisateur QMQM et QMQMADM disposent des droits nécessaires et QMQM possède des mots de passe correspondants sur les trois systèmes.

2. Ajoutez des entrées de base de données relationnelle (RDBE) aux systèmes IBM i qui vont héberger des instances de gestionnaire de files d'attente.
  - a) Sur ALPHA, créez la connexion à BETA.

```
ADDRDBDIRE RDB(BETA) RMTLOCNAME(BETA *IP) RMTAUTMTH(*USRIDPWD)
```

- b) Sur BETA, créez les connexions à ALPHA.

```
ADDRDBDIRE RDB(ALPHA) RMTLOCNAME(ALPHA *IP) RMTAUTMTH(*USRIDPWD)
```

3. Créez les scripts qui recréent les objets de gestionnaire de files d'attente.

```
QSAVEQMGR LCLQMGRNAM(QM1) FILENAME('*CURLIB/QMQSC(QM1)')  
OUTPUT(*REPLACE) MAKEAUTH(*YES) AUTHFN('*CURLIB/QMAUT(QM1)')
```

4. Arrêtez le gestionnaire de files d'attente et supprimez-le.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ENDCCTJOB(*YES) RCDMQMIMG(*YES) TIMEOUT(15)  
DLTMQM MQMNAME(QM1)
```

5. Créez le gestionnaire de files d'attente QM1 sur ALPHA, en sauvegardant les données du gestionnaire de files d'attente sur GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)  
MQMDIRP('/QNTC/GAMMA/WMQ')
```

Le chemin d'accès utilise NetServer pour créer les données du gestionnaire de files d'attente.

6. Recréez les objets de gestionnaire de files d'attente pour QM1 à partir des définitions sauvegardées.

```
STRMQMQSC SRCMBR(QM1) SRCFILE(*CURLIB/QMQSC) MQMNAME(QM1)
```

7. Appliquez les autorisations à partir des informations sauvegardées.

- a) Compilez le programme d'autorisation sauvegardé.

```
CRTCLPGM PGM(*CURLIB/QM1) SRCFILE(*CURLIB/QMAUT)
SRCMBR(QM1) REPLACE(*YES)
```

- b) Exécutez le programme pour appliquer les autorisations.

```
CALL PGM(*CURLIB/QM1)
```

- c) Actualisez les informations de sécurité pour QM1.

```
RFRMQMAUT MQMNAME(QM1)
```

8. Exécuter sur ALPHA. La commande ajoute un journal éloigné sur BETA pour.

```
ADDQMJRNM MQMNAME(QM1) RMTJRNRDB(BETA)
```

crée des entrées de journal dans son journal local sur ALPHA lorsque l'instance active est sur ALPHA. Le journal local sur ALPHA est répliqué sur le journal éloigné sur BETA.

9. Utilisez la commande, pour inspecter les données de configuration IBM MQ créées par pour sur ALPHA.

Les informations sont nécessaires à l'étape suivante.

Dans cet exemple, la configuration suivante est créée dans ALPHA pour:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

10. Créez une instance de gestionnaire de files d'attente de QM1 sur BETA à l'aide de la commande. Exécutez la commande suivante sur BETA pour modifier les informations de contrôle du gestionnaire de files d'attente sur BETA.

```
ADDQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

**Conseil :** Copiez et collez les informations de configuration. La section du gestionnaire de files d'attente est la même sur ALPHA et BETA.

11. Exécuter sur BETA. La commande ajoute un journal local sur BETA et un journal éloigné sur ALPHA pour.

```
ADDQMJRNM MQMNAME(QM1) RMTJRNRDB(ALPHA)
```

crée des entrées de journal dans son journal local sur BETA lorsque l'instance active est sur BETA. Le journal local sur BETA est répliqué sur le journal éloigné sur ALPHA.

**Remarque :** Vous pouvez également configurer la journalisation à distance de BETA à ALPHA à l'aide de la journalisation asynchrone.

Utilisez cette commande pour configurer la journalisation asynchrone de BETA vers ALPHA, à la place de la commande de l'étape «7», à la page 447.

```
ADDQMJRNM MQMNAME(QM1) RMTJRNRDB(ALPHA) RMTJRNDLV(*ASYNCR)
```

Si le serveur ou la journalisation sur ALPHA est la source de l'échec, la fonction BETA démarre sans attendre que de nouvelles entrées de journal soient répliquées sur ALPHA.

Basculez le mode de répllication sur \*SYNC, à l'aide de la commande, lorsque ALPHA est à nouveau en ligne.

Utilisez les informations de la rubrique «[Configuration de journal protégé par disque miroir pour ASP sous IBM i](#)», à la page 440 pour déterminer si les journaux doivent être mis en miroir de manière synchrone, asynchrone ou mixte. Par défaut, la répllication est synchrone, avec une période d'attente de 60 secondes pour une réponse du journal éloigné.

12. Vérifiez que les journaux ALPHA et BETA sont activés et que le statut de la répllication de journal éloigné est activé.

- a) Sur ALPHA:

```
WRKMQMJRN MQMNAME(QM1)
```

- b) Sur BETA:

```
WRKMQMJRN MQMNAME(QM1)
```

13. Démarrez les instances de gestionnaire de files d'attente sur ALPHA et BETA.

- a) Démarrez la première instance sur ALPHA, ce qui en fait l'instance active. Activation du basculement vers une instance de secours.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) Démarrez la deuxième instance sur BETA, ce qui en fait l'instance de secours.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

## Résultats

Permet de vérifier le statut du gestionnaire de files d'attente:

1. Le statut de l'instance de gestionnaire de files d'attente sur ALPHA doit être.
2. Le statut de l'instance de gestionnaire de files d'attente sur BETA doit être.

## Exemple

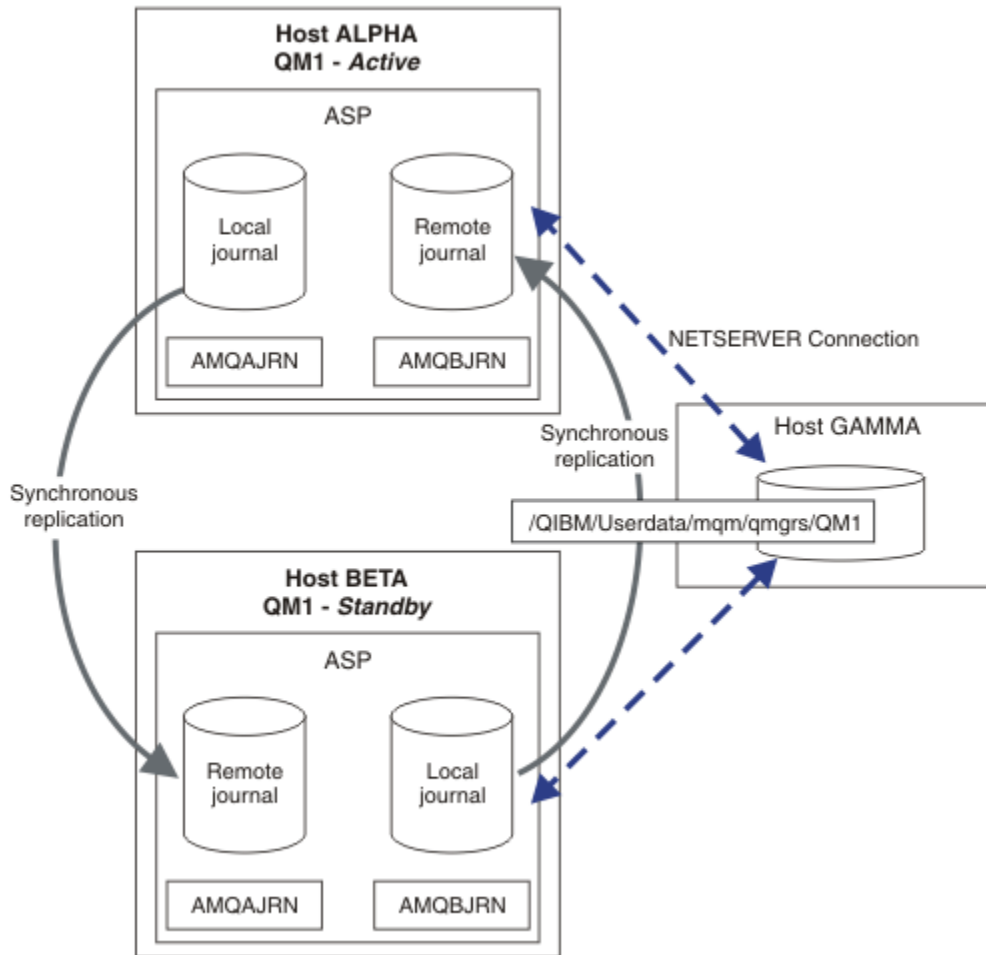


Figure 26. Configuration du journal en miroir

## Que faire ensuite

- Vérifiez que les instances actives et de secours basculent automatiquement. Vous pouvez exécuter les exemples de programmes à haute disponibilité pour tester le basculement ; voir [Exemples de programmes à haute disponibilité](#). Les exemples de programme sont des clients 'C'. Vous pouvez les exécuter à partir d'une plateforme Windows ou Unix.

1. Démarrez les exemples de programmes à haute disponibilité.
2. Sous ALPHA, arrêtez le gestionnaire de files d'attente demandant le basculement:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Vérifiez que l'instance de sur BETA est active.
4. Redémarrer sur ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Examinez les autres configurations à haute disponibilité:
  1. Utilisez NetServer pour placer les données du gestionnaire de files d'attente sur un serveur Windows .

2. Au lieu d'utiliser la journalisation éloignée pour reproduire le journal du gestionnaire de files d'attente, stockez le journal sur un ASP indépendant. Utilisez la mise en grappe IBM i pour transférer l'ASP indépendant de ALPHA vers BETA.

### **IBM i Configuration de journal ASP indépendant commuté sous IBM i**

Il n'est pas nécessaire de répliquer un journal ASP indépendant pour créer une configuration de gestionnaire de files d'attente multi-instance. Vous devez automatiser un moyen de transférer l'ASP indépendant du gestionnaire de files d'attente actif vers le gestionnaire de files d'attente de secours. Il existe d'autres solutions de haute disponibilité possibles à l'aide d'un ASP indépendant, qui ne nécessitent pas toutes l'utilisation d'un gestionnaire de files d'attente multi-instance.

Lorsque vous utilisez un ASP indépendant, vous n'avez pas besoin de mettre en miroir le journal du gestionnaire de files d'attente. Si vous avez installé la gestion de cluster et que les serveurs hébergeant les instances de gestionnaire de files d'attente se trouvent dans le même groupe de ressources en cluster, le journal du gestionnaire de files d'attente peut être transféré automatiquement à un autre serveur à une courte distance du serveur actif, si l'hôte exécutant l'instance active échoue. Vous pouvez également transférer le journal manuellement, dans le cadre d'un commutateur planifié, ou vous pouvez écrire une procédure de commande pour transférer l'ASP indépendant à l'aide d'un programme.

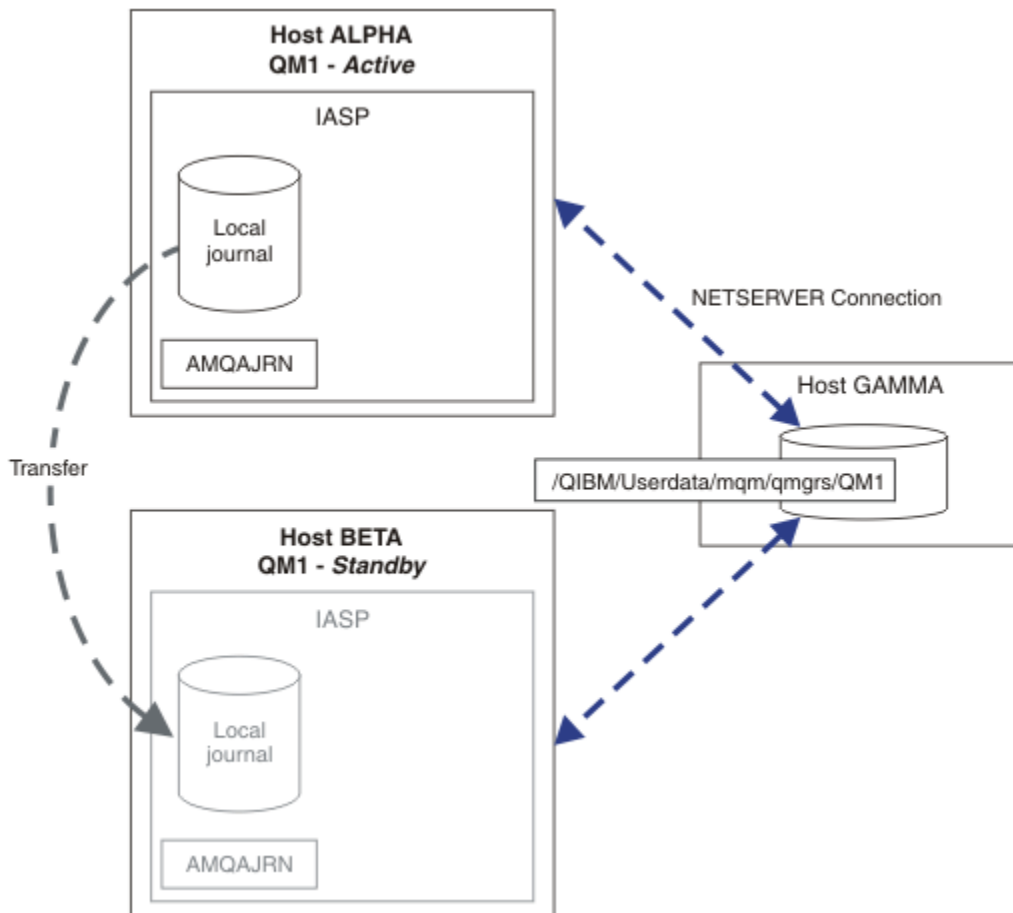


Figure 27. Transfert d'un journal de gestionnaire de files d'attente à l'aide d'un ASP indépendant

Pour le fonctionnement d'un gestionnaire de files d'attente multi-instance, les données du gestionnaire de files d'attente doivent être stockées sur un système de fichiers partagé. Le système de fichiers peut être hébergé sur différentes plateformes. Vous ne pouvez pas stocker de données de gestionnaire de files d'attente multi-instance sur un ASP ou un ASP indépendant.

Le système de fichiers partagé joue deux rôles dans la configuration: les mêmes données de gestionnaire de files d'attente sont partagées entre toutes les instances du gestionnaire de files d'attente. Le système


de fichiers doit disposer d'un protocole de verrouillage robuste qui garantit qu'une seule instance du gestionnaire de files d'attente a accès aux données du gestionnaire de files d'attente une fois qu'elle a démarré. Si le gestionnaire de files d'attente est défaillant ou que les communications avec le serveur de fichiers sont rompées, le système de fichiers doit libérer le verrou sur les données du gestionnaire de files d'attente détenues par l'instance active qui ne communique plus avec le système de fichiers. L'instance de gestionnaire de files d'attente de secours peut alors accéder en lecture / écriture aux données du gestionnaire de files d'attente. Le protocole de système de fichiers doit être conforme à un ensemble de règles pour fonctionner correctement avec les gestionnaires de files d'attente multi-instance (voir «Composants d'une solution à haute disponibilité sur IBM i», à la page 431).

Le mécanisme de verrouillage sérialise la commande de démarrage du gestionnaire de files d'attente et contrôle l'instance du gestionnaire de files d'attente qui est active. Une fois qu'un gestionnaire de files d'attente devient actif, il reconstruit ses files d'attente à partir du journal local que vous ou le cluster à haute disponibilité avez transféré vers le serveur de secours. Les clients reconnectables en attente de reconnexion au même gestionnaire de files d'attente sont reconnectés et toutes les transactions en cours sont annulées. Les applications configurées pour démarrer en tant que services de gestionnaire de files d'attente sont démarrées.

Vous devez vous assurer que le journal local de l'instance de gestionnaire de files d'attente active qui a échoué sur l'ASP indépendant est transféré sur le serveur qui héberge l'instance de gestionnaire de files d'attente de secours qui vient d'être activée, soit en configurant le gestionnaire de ressources en grappe, soit en transférant manuellement l'ASP indépendant. L'utilisation d'ASP indépendants n'empêche pas la configuration de journaux éloignés et de la mise en miroir, si vous décidez d'utiliser un ASP indépendant pour la sauvegarde et la reprise après incident, et d'utiliser la mise en miroir de journaux éloignés pour la configuration du gestionnaire de files d'attente multi-instance.

Si vous avez choisi d'utiliser un ASP indépendant, vous pouvez envisager d'autres configurations à haute disponibilité. L'arrière-plan de ces solutions est décrit dans «ASP indépendants et haute disponibilité», à la page 459.

1. Au lieu d'utiliser des gestionnaires de files d'attente multi-instance, installez et configurez un gestionnaire de files d'attente à instance unique entièrement sur un ASP indépendant, et utilisez les services de haute disponibilité IBM i pour faire basculer le gestionnaire de files d'attente. Vous devrez probablement augmenter la solution avec un moniteur de gestionnaire de files d'attente pour détecter si le gestionnaire de files d'attente a échoué indépendamment du serveur. Il s'agit de la base de la solution fournie dans *Supportpac MC41: Configuration de IBM MQ for iSeries for High Availability*.
2. Utilisez des ASP indépendants et la protection par disque miroir multisite (XSM) pour mettre en miroir l'ASP indépendant au lieu de le permuter sur le bus local. Cela étend la plage géographique de la solution ASP indépendante jusqu'à ce que le temps nécessaire à l'écriture des enregistrements de journal sur une longue distance le permette.

 *Création d'un gestionnaire de files d'attente multi-instance à l'aide d'un ASP indépendant et de NetServer sous IBM i*

Créez un gestionnaire de files d'attente multi-instance à exécuter sur deux serveurs IBM i . Les données du gestionnaire de files d'attente sont stockées sur un serveur IBM i à l'aide de NetServer. Le journal du gestionnaire de files d'attente est stocké dans un ASP indépendant. Utilisez la mise en grappe IBM i ou une procédure manuelle pour transférer l'ASP indépendant contenant le journal du gestionnaire de files d'attente vers l'autre serveur IBM i .

## Avant de commencer

1. La tâche requiert trois serveurs IBM i . Installez IBM MQ sur deux d'entre eux, ALPHA et BETA dans l'exemple. le produit doit être au moins au niveau IBM WebSphere MQ 7.0.1 Fix Pack 1.
2. Le troisième serveur est un serveur IBM i , connecté par NetServer à ALPHA et BETA. Il est utilisé pour partager les données du gestionnaire de files d'attente. Il n'est pas nécessaire d'avoir une installation IBM MQ . Il est utile d'installer IBM MQ sur le serveur en tant qu'étape temporaire pour configurer les répertoires et les droits du gestionnaire de files d'attente.
3. Assurez-vous que le profil utilisateur QMQM possède le même mot de passe sur les trois serveurs.



4. Installez IBM i NetServer; voir [i5/OS NetServer](#).
5. Créez des procédures pour transférer l'ASP indépendant du gestionnaire de files d'attente défaillant vers le serveur de secours qui prend le relais. Certaines des techniques décrites dans *SupportPac MC41: Configuration de IBM MQ for iSeries for High Availability* peuvent être utiles lors de la conception de vos procédures de transfert ASP indépendant.

## Pourquoi et quand exécuter cette tâche

Procédez comme suit pour créer la configuration illustrée dans la [Figure 28](#), à la page 458. Les données du gestionnaire de files d'attente sont connectées à l'aide d' IBM i NetServer.

- Créez des connexions depuis ALPHA et BETA vers le partage de répertoire sur GAMMA qui permet de stocker les données du gestionnaire de files d'attente. La tâche configure également les droits d'accès, les profils utilisateur et les mots de passe nécessaires.
- Créez le gestionnaire de files d'attente QM1 sur le serveur IBM i , ALPHA.
- Ajoutez les informations de contrôle du gestionnaire de files d'attente pour QM1 sur l'autre serveur IBM i , BETA.
- Démarrez le gestionnaire de files d'attente en autorisant une instance de secours.

## Procédure

1. Exécutez la tâche [«Création d'un partage de réseau pour les données de gestionnaire de files d'attente à l'aide de NetServer sous IBM i»](#), à la page 432.

Par conséquent, ALPHA et BETA ont un partage, /QNTC/GAMMA/WMQ, qui pointe vers /QIBM/UserData/mqm/qmgrs sur GAMMA. Les profils utilisateur QMQM et QMQMADM disposent des droits nécessaires et QMQM possède des mots de passe correspondants sur les trois systèmes.

2. Créez le gestionnaire de files d'attente QM1 sur ALPHA, en sauvegardant les données du gestionnaire de files d'attente sur GAMMA.

```
CRTMQM MQMNAME(QM1) UDLMSGQ(SYSTEM.DEAD.LETTER.QUEUE)
MQMDIRP(' /QNTC/GAMMA/WMQ')
```

Le chemin d'accès utilise NetServer pour créer les données du gestionnaire de files d'attente.

3. Utilisez la commande, pour inspecter les données de configuration IBM MQ créées par pour sur ALPHA.

Les informations sont nécessaires à l'étape suivante.

Dans cet exemple, la configuration suivante est créée dans ALPHA pour:

```
Name=QM1
Prefix=/QIBM/UserData/mqm
Library=QMOM1
Directory=QM1
DataPath= /QNTC/GAMMA/WMQ /QM1
```

4. Créez une instance de gestionnaire de files d'attente de QM1 sur BETA à l'aide de la commande. Exécutez la commande suivante sur BETA pour modifier les informations de contrôle du gestionnaire de files d'attente sur BETA.

```
ADDMQMINF MQMNAME(QM1)
PREFIX('/QIBM/UserData/mqm')
MQMDIR(QM1)
MQMLIB(QMOM1)
DATAPATH('/QNTC/GAMMA/WMQ /QM1')
```

**Conseil :** Copiez et collez les informations de configuration. La section du gestionnaire de files d'attente est la même sur ALPHA et BETA.

5. Démarrez les instances de gestionnaire de files d'attente sur ALPHA et BETA.

- a) Démarrez la première instance sur ALPHA, ce qui en fait l'instance active. Activation du basculement vers une instance de secours.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- b) Démarrez la deuxième instance sur BETA, ce qui en fait l'instance de secours.

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

## Résultats

Permet de vérifier le statut du gestionnaire de files d'attente:

1. Le statut de l'instance de gestionnaire de files d'attente sur ALPHA doit être.
2. Le statut de l'instance de gestionnaire de files d'attente sur BETA doit être.

## Exemple

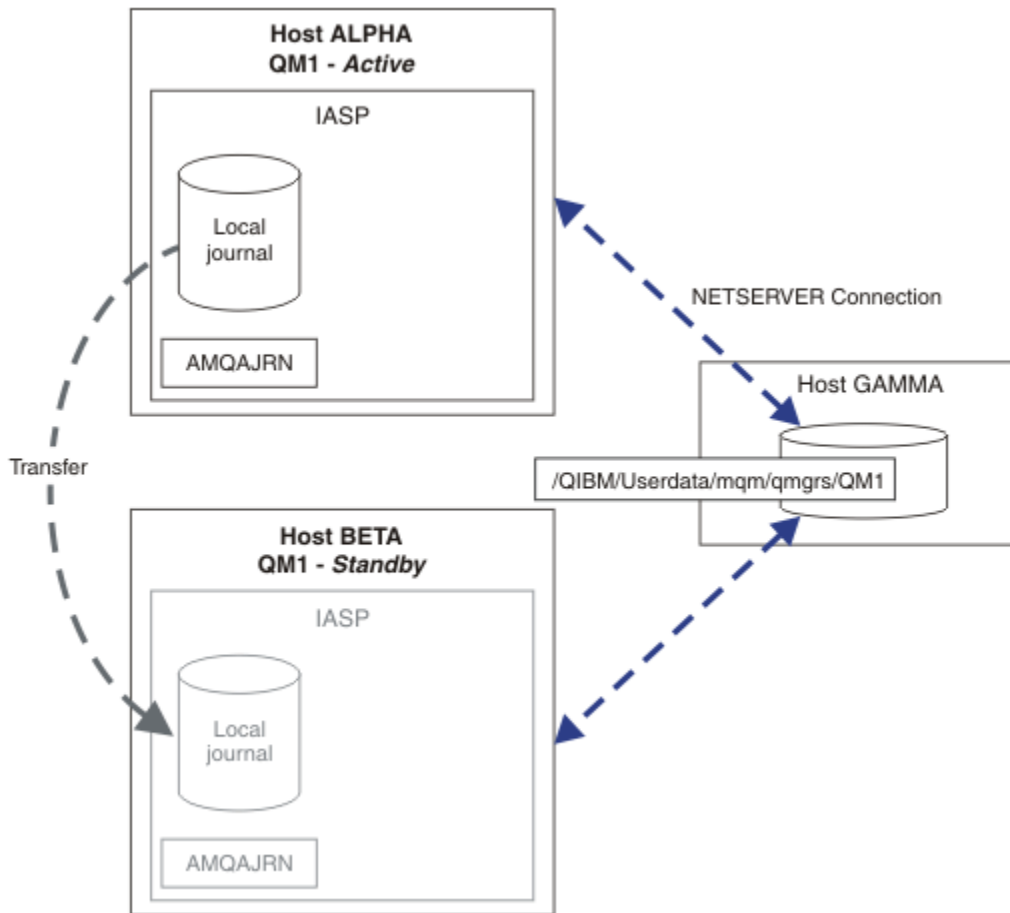


Figure 28. Transfert d'un journal de gestionnaire de files d'attente à l'aide d'un ASP indépendant

## Que faire ensuite

- Vérifiez que les instances actives et de secours basculent automatiquement. Vous pouvez exécuter les exemples de programmes à haute disponibilité pour tester le basculement ; voir [Exemples de](#)

programmes à haute disponibilité. Les exemples de programme sont des clients 'C'. Vous pouvez les exécuter à partir d'une plateforme Windows ou Unix.

1. Démarrez les exemples de programmes à haute disponibilité.
2. Sous ALPHA, arrêtez le gestionnaire de files d'attente demandant le basculement:

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) ALSWITCH(*YES)
```

3. Vérifiez que l'instance de sur BETA est active.
4. Redémarrer sur ALPHA

```
STRMQM MQMNAME(QM1) STANDBY(*YES)
```

- Examinez les autres configurations à haute disponibilité:
  1. Utilisez NetServer pour placer les données du gestionnaire de files d'attente sur un serveur IBM i .
  2. Au lieu d'utiliser un ASP indépendant pour transférer le journal du gestionnaire de files d'attente sur le serveur de secours, utilisez la journalisation à distance pour reproduire le journal sur le serveur de secours.

### IBM i ASP indépendants et haute disponibilité

Les ASP indépendants permettent de déplacer des applications et des données entre les serveurs. La flexibilité des ASP indépendants signifie qu'ils constituent la base de certaines solutions à haute disponibilité IBM i . Lorsque vous envisagez d'utiliser un ASP ou un ASP indépendant pour le journal du gestionnaire de files d'attente, vous devez prendre en compte une autre configuration à haute disponibilité basée sur des ASP indépendants.

Les pools de mémoire secondaire (ASP) sont un bloc de construction de l'architecture IBM i . Les unités de disques sont regroupées pour former un ASP unique. En plaçant des objets dans différents ASP, vous pouvez protéger les données d'un ASP contre les défaillances de disque dans un autre ASP.

Chaque serveur IBM i possède au moins un ASP *de base* , appelé ASP système. Il est désigné comme ASP1 et parfois appelé \*SYSBAS. Vous pouvez configurer jusqu'à 31 ASP *utilisateur* de base supplémentaires qui ne peuvent pas être distingués de l'ASP système du point de vue de l'application, car ils partagent le même espace de nom. En utilisant plusieurs ASP de base pour distribuer des applications sur de nombreux disques, vous pouvez améliorer les performances et réduire le temps de reprise. L'utilisation de plusieurs ASP de base peut également fournir un certain degré d'isolement en cas de panne de disque, mais elle n'améliore pas la fiabilité globale.

Les ASP indépendants sont un type spécial d'ASP. Ils sont souvent appelés pools de stockage sur disque indépendants. Les pools de stockage sur disque indépendants sont des composants clés de la haute disponibilité IBM i . Vous pouvez stocker des données et des applications qui se considèrent comme indépendantes du système en cours auquel elles sont connectées sur des unités de stockage sur disque indépendantes. Vous pouvez configurer des ASP indépendants commutables ou non commutables. Du point de vue de la disponibilité, vous ne vous souciez généralement que des ASP indépendants commutables, qui peuvent être transférés automatiquement d'un serveur à un autre. Par conséquent, vous pouvez déplacer les applications et les données de l'ASP indépendant d'un serveur à un autre.

Contrairement aux ASP utilisateur de base, les ASP indépendants ne partagent pas le même espace de nom que l'ASP système. Les applications qui utilisent des ASP utilisateur nécessitent des modifications pour fonctionner avec un ASP indépendant. Vous devez vérifier que votre logiciel et le logiciel tiers que vous utilisez fonctionnent dans un environnement ASP indépendant.

Lorsque l'ASP indépendant est connecté à un autre serveur, l'espace de nom de l'ASP indépendant doit être combiné avec l'espace de nom de l'ASP système. Ce processus est appelé *mise en fonction* de l'ASP indépendant. Vous pouvez mettre en fonction un ASP indépendant sans effectuer d'IPL du serveur. La prise en charge de la mise en grappe est requise pour le transfert automatique des ASP indépendants d'un serveur à un autre.

## Création de solutions fiables avec des ASP indépendants

La journalisation dans un ASP indépendant, plutôt que dans un ASP et à l'aide de la réplication de journal, offre un autre moyen de fournir au gestionnaire de files d'attente de secours une copie du journal local à partir de l'instance de gestionnaire de files d'attente défaillante. Pour transférer automatiquement l'ASP indépendant vers un autre serveur, vous devez avoir installé et configuré la prise en charge de la mise en grappe. Il existe un certain nombre de solutions à haute disponibilité pour les ASP indépendants en fonction de la prise en charge du cluster et de la mise en miroir de disque de bas niveau, que vous pouvez combiner avec des gestionnaires de files d'attente multi-instance ou les remplacer.

La liste suivante décrit les composants nécessaires à la génération d'une solution fiable basée sur des ASP indépendants.

### Journalisation

Les gestionnaires de files d'attente et d'autres applications utilisent des journaux locaux pour écrire des données persistantes en toute sécurité sur le disque afin de les protéger contre la perte de données en mémoire en cas de défaillance du serveur. Cette cohérence est parfois appelée cohérence ponctuelle. Elle ne garantit pas la cohérence des mises à jour multiples qui ont lieu sur une période donnée.

### Commitment Control

En utilisant des transactions globales, vous pouvez coordonner les mises à jour des messages et des bases de données afin que les données écrites dans le journal soient cohérentes. Il assure la cohérence sur une période de temps en utilisant un protocole de validation en deux phases.

### Disque commuté

Les disques commutés sont gérés par le groupe de ressources en grappe d'unité dans une grappe à haute disponibilité. Le groupe de ressources en grappe bascule automatiquement les ASP indépendants vers un nouveau serveur en cas d'indisponibilité non planifiée. Les groupes de ressources en grappe sont géographiquement limités à l'étendue du bus d'E-S local.

En configurant votre journal local sur un ASP indépendant commutable, vous pouvez transférer le journal sur un autre serveur et reprendre le traitement des messages. Aucune modification apportée aux messages persistants sans contrôle de point de synchronisation ou validés avec contrôle de point de synchronisation n'est perdue, sauf si l'ASP indépendant est défaillant.

Si vous utilisez à la fois la journalisation et le contrôle de validation sur des ASP indépendants commutables, vous pouvez transférer des journaux de base de données et des journaux de gestionnaire de files d'attente vers un autre serveur et reprendre le traitement des transactions sans perte de cohérence ni de transactions validées.

### Mise en miroir intersite (XSM)

XSM met en miroir l'ASP indépendant principal vers un ASP indépendant secondaire géographiquement éloigné sur un réseau TCP/IP et transfère automatiquement le contrôle en cas de défaillance. Vous avez le choix entre la configuration d'un miroir synchrone ou asynchrone. La mise en miroir synchrone réduit les performances du gestionnaire de files d'attente car les données sont mises en miroir avant la fin des opérations d'écriture sur le système de production, mais elle garantit que l'ASP indépendant secondaire est à jour. Si vous utilisez la mise en miroir asynchrone, vous ne pouvez pas garantir que l'ASP indépendant secondaire est à jour. La mise en miroir asynchrone assure la cohérence de l'ASP indépendant secondaire.

Il existe trois technologies XSM.

### Protection géographique par disque miroir

La protection géographique par disque miroir est une extension de la mise en grappe, qui vous permet de basculer entre les ASP indépendants sur une vaste zone. Il possède à la fois des modes synchrone et asynchrone. Vous pouvez garantir la haute disponibilité uniquement en mode synchrone, mais la séparation des ASP indépendants peut avoir un impact trop important sur les performances. Vous pouvez combiner la protection géographique par disque miroir avec un disque commuté pour assurer la haute disponibilité en local et la reprise après incident à distance.

### **Mise en miroir Metro**

La mise en miroir Metro est un service de niveau unité qui fournit une mise en miroir synchrone locale rapide sur des distances plus longues que le bus local. Vous pouvez le combiner avec un gestionnaire de files d'attente multi-instance pour vous assurer de la haute disponibilité du gestionnaire de files d'attente et en disposant de deux copies de l'ASP indépendant, de la haute disponibilité du journal du gestionnaire de files d'attente.

### **Mise en miroir globale**

La mise en miroir globale est un service de niveau unité qui fournit une mise en miroir asynchrone et qui convient à la sauvegarde et à la reprise après incident sur des distances plus longues, mais qui n'est pas un choix normal pour la haute disponibilité, car il ne maintient que la cohérence du point de cohérence plutôt que la devise.

Les principaux points de décision que vous devez prendre en considération sont les suivants:

#### **ASP ou ASP indépendant?**

Il n'est pas nécessaire d'exécuter un cluster à haute disponibilité IBM i pour utiliser des gestionnaires de files d'attente multi-instance. Vous pouvez choisir des ASP indépendants, si vous utilisez déjà des ASP indépendants, ou si vous avez des exigences de disponibilité pour d'autres applications qui requièrent des ASP indépendants. Il peut être utile de combiner des ASP indépendants avec des gestionnaires de files d'attente multi-instance pour remplacer la surveillance des gestionnaires de files d'attente afin de détecter les incidents de gestionnaire de files d'attente.

#### **Disponibilité?**

Quel est l'objectif de temps de reprise (RTO)? Si vous avez besoin de l'apparence d'un comportement quasi ininterrompu, quelle solution a le temps de récupération le plus rapide?

#### **Disponibilité du journal?**

Comment éliminer le journal en tant que point de défaillance unique. Vous pouvez adopter une solution matérielle, à l'aide de périphériques RAID 1 ou d'une version supérieure, ou vous pouvez combiner ou utiliser une solution logicielle à l'aide de journaux de réplique ou de la mise en miroir de disque.

#### **Distance?**

À quelle distance se trouvent les instances de gestionnaire de files d'attente actives et de secours. Vos utilisateurs peuvent-ils tolérer la dégradation des performances de la réplique synchrone sur des distances supérieures à environ 250 mètres?

#### **Compétences?**

Il y a du travail à faire pour automatiser les tâches administratives liées à la gestion et à l'exercice réguliers de la solution. Les compétences requises pour effectuer l'automatisation sont différentes pour les solutions basées sur les ASP et les ASP indépendants.

## **Suppression d'un gestionnaire de files d'attente multi-instance sous IBM i**

Avant de supprimer un gestionnaire de files d'attente multi-instance, arrêtez la journalisation à distance et supprimez des instances de gestionnaire de files d'attente.

### **Avant de commencer**

1. Dans cet exemple, deux instances du gestionnaire de files d'attente QM1 sont définies sur les serveurs ALPHA et BETA. ALPHA est l'instance active et BETA est l'instance de secours. Les données du gestionnaire de files d'attente associées au gestionnaire de files d'attente QM1 sont stockées sur le serveur IBM i GAMMA, à l'aide de NetServer. Voir [«Création d'un gestionnaire de files d'attente multi-instance à l'aide de la mise en miroir du journal et de NetServer sous IBM i»](#), à la page 446.
2. ALPHA et BETA doivent être connectés pour que les journaux éloignés définis puissent être supprimés par IBM MQ.
3. Vérifiez que le répertoire /QNTC et le partage de fichiers du répertoire du serveur sont accessibles à l'aide des commandes système **EDTF** ou **WRKLNK**

## Pourquoi et quand exécuter cette tâche

Avant de supprimer un gestionnaire de files d'attente multi-instance d'un serveur à l'aide de la commande **DLTMQM**, supprimez toutes les instances de gestionnaire de files d'attente sur d'autres serveurs à l'aide de la commande **RMVMQMINF**.

Lorsque vous supprimez une instance de gestionnaire de files d'attente à l'aide de la commande **RMVMQMINF**, les journaux locaux et distants préfixés par AMQet associés à l'instance sont supprimés. Les informations de configuration relatives à l'instance de gestionnaire de files d'attente, locale sur le serveur, sont également supprimées.

N'exécutez pas la commande **RMVMQMINF** sur le serveur contenant l'instance restante du gestionnaire de files d'attente. Cela empêche **DLTMQM** de fonctionner correctement.

Supprimez le gestionnaire de files d'attente à l'aide de la commande **DLTMQM**. Les données du gestionnaire de files d'attente sont supprimées du partage de réseau. Les journaux locaux et éloignés préfixés avec AMQ et associés à l'instance sont supprimés. **DLTMQM** supprime également les informations de configuration relatives à l'instance de gestionnaire de files d'attente, locale sur le serveur.

Dans l'exemple, il n'y a que deux instances de gestionnaire de files d'attente. IBM MQ prend en charge une configuration multi-instance en cours d'exécution qui comporte une instance de gestionnaire de files d'attente active et une instance de secours. Si vous avez créé des instances de gestionnaire de files d'attente supplémentaires à utiliser dans des configurations en cours d'exécution, supprimez-les à l'aide de la commande **RMVMQMINF** avant de supprimer l'instance restante.

## Procédure

1. Exécutez la commande **CHGMQMJRN RMTJRNSTS (\*INACTIVE)** sur chaque serveur pour rendre la journalisation à distance entre les instances de gestionnaire de files d'attente inactive.

a) Sur ALPHA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('BETA') RMTJRNSTS(*INACTIVE)
```

b) Sur BETA:

```
CHGMQMJRN MQMNAME('QM1')
RMTJRNRDB('ALPHA') RMTJRNSTS(*INACTIVE)
```

2. Exécutez la commande **ENDMQM** sur ALPHA, l'instance de gestionnaire de files d'attente active, pour arrêter les deux instances de QM1.

```
ENDMQM MQMNAME(QM1) OPTION(*IMMED) INSTANCE(*ALL) ENDCCTJOB(*YES)
```

3. Exécutez la commande **RMVMQMINF** sur ALPHA pour supprimer les ressources de gestionnaire de files d'attente de l'instance de ALPHA et de BETA.

```
RMVMQMINF MQMNAME(QM1)
```

**RMVMQMINF** supprime les informations de configuration du gestionnaire de files d'attente pour QM1 de ALPHA. Si le nom de journal est préfixé par AMQ, il supprime le journal local associé à QM1 de ALPHA. Si le nom de journal est préfixé par AMQ et qu'un journal éloigné a été créé, il supprime également le journal éloigné de BETA.

4. Exécutez la commande **DLTMQM** sur BETA pour supprimer QM1.

```
DLTMQM MQMNAME(QM1)
```

**DLTMQM** supprime les données du gestionnaire de files d'attente du partage de réseau sur GAMMA. Elle supprime les informations de configuration du gestionnaire de files d'attente pour QM1 de BETA.

Si le nom du journal est précédé du préfixe AMQ, le journal local associé à QM1 est supprimé de BETA. Si le nom du journal est préfixé par AMQ et qu'un journal éloigné a été créé, il supprime également le journal éloigné de ALPHA.

## Résultats

**DLTMQM** et **RMVMQMINF** suppriment les journaux locaux et éloignés créés par **CRTMQM** et **ADDMQJRN**. Les commandes suppriment également les récepteurs de journal. Les journaux et les récepteurs de journal doivent respecter la convention de dénomination selon laquelle les noms doivent commencer par AMQ. **DLTMQM** et **RMVMQMINF** suppriment les objets du gestionnaire de files d'attente, les données du gestionnaire de files d'attente et les informations de configuration du gestionnaire de files d'attente de `mqs.ini`.

## Que faire ensuite

Une autre approche consiste à émettre les commandes suivantes après la désactivation de la journalisation à l'étape «1», à la page 462 et avant l'arrêt des instances de gestionnaire de files d'attente. Sinon, si vous n'avez pas suivi la convention de dénomination, vous devez supprimer les journaux et les récepteurs de journal par leur nom.

1. Sur ALPHA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('BETA')
```

2. Sur BETA:

```
RMVMQMJRN MQMNAME('QM1') RMTJRNRDB('ALPHA')
```

Après avoir supprimé les journaux, passez au reste des étapes.

## **Sauvegarde d'un gestionnaire de files d'attente multi-instance sous IBM i**

Cette procédure explique comment sauvegarder les objets de gestionnaire de files d'attente sur le serveur local et les données du gestionnaire de files d'attente sur le serveur de fichiers réseau. Adaptez l'exemple pour sauvegarder des données pour d'autres gestionnaires de files d'attente.

## Avant de commencer

Dans cet exemple, les données de gestionnaire de files d'attente associées au gestionnaire de files d'attente QM1 sont stockées sur le serveur IBM i appelé GAMMA, à l'aide de NetServer. Voir «Création d'un gestionnaire de files d'attente multi-instance à l'aide de la mise en miroir du journal et de NetServer sous IBM i», à la page 446. IBM MQ est installé sur les serveurs, ALPHA et BETA. Le gestionnaire de files d'attente, QM1, est configuré sur ALPHA et BETA.

## Pourquoi et quand exécuter cette tâche

IBM i ne prend pas en charge la sauvegarde des données à partir d'un répertoire distant. Sauvegardez les données du gestionnaire de files d'attente sur un système de fichiers distant à l'aide des procédures de sauvegarde locales sur le serveur de système de fichiers. Dans cette tâche, le système de fichiers réseau se trouve sur un serveur IBM i, GAMMA. Les données du gestionnaire de files d'attente sont sauvegardées dans un fichier sauvegarde sur GAMMA.

Si le système de fichiers réseau se trouve sous Windows ou Linux, vous pouvez stocker les données du gestionnaire de files d'attente dans un fichier compressé, puis les sauvegarder. Si vous disposez d'un système de sauvegarde, tel que Tivoli Storage Manager, utilisez-le pour sauvegarder les données du gestionnaire de files d'attente.

## Procédure

1. Créez un fichier sauvegarde sur ALPHA pour la bibliothèque de gestionnaire de files d'attente associée à QM1.

Utilisez le nom de la bibliothèque du gestionnaire de files d'attente pour nommer le fichier sauvegarde.

```
CRTSAVF FILE(QGPL/QMQM1)
```

2. Sauvegardez la bibliothèque du gestionnaire de files d'attente dans le fichier sauvegarde sur ALPHA.

```
SAVLIB LIB(QMQM1) DEV(*SAVF) SAVF(QGPL/QMQM1)
```

3. Créez un fichier sauvegarde pour le répertoire de données du gestionnaire de files d'attente sur GAMMA.

Utilisez le nom du gestionnaire de files d'attente pour nommer le fichier sauvegarde.

```
CRTSAVF FILE(QGPL/QMDQM1)
```

4. Sauvegardez la copie des données du gestionnaire de files d'attente à partir du répertoire local sur GAMMA.

```
SAV DEV('/QSYS.LIB/QGPL.LIB/QMDQM1.FILE') OBJ('/QIBM/Userdata/mqm/qmgrs/QM1')
```

## **Commandes de configuration de gestionnaires de files d'attente multi-instance**

IBM MQ comporte des commandes permettant de simplifier la configuration de la réplication de journal, d'ajouter de nouvelles instances de gestionnaire de files d'attente et de configurer les gestionnaires de files d'attente pour utiliser un ASP indépendant.

Les commandes de journal permettant de créer et de gérer des journaux locaux et éloignés sont les suivantes:

### **ADDMQMJRN**

Avec cette commande, vous pouvez créer des journaux locaux et éloignés nommés pour une instance de gestionnaire de files d'attente et configurer si la réplication est synchrone ou asynchrone, en quoi consiste le délai d'attente synchrone et si le journal éloigné doit être activé immédiatement.

### **CHGMQMJRN**

La commande modifie les paramètres de délai d'attente, de statut et de distribution affectant les journaux de réplique.

### **RMVMQMJRN**

Supprime les journaux *distant*s nommés d'une instance de gestionnaire de files d'attente.

### **WRKMQMJRN**

Répertorie l'état des journaux locaux et éloignés pour une instance de gestionnaire de files d'attente locale.

Ajoutez et gérez des instances de gestionnaire de files d'attente supplémentaires à l'aide des commandes suivantes, qui modifient le fichier `mqs.ini`.

### **ADDMQMINF**

La commande utilise les informations que vous avez extraites du fichier `mqs.ini` à l'aide de la commande `DSPMQMINF` pour ajouter une nouvelle instance de gestionnaire de files d'attente sur un autre serveur IBM i.

### **RMVMQMINF**

Supprimez une instance de gestionnaire de files d'attente. Utilisez cette commande pour supprimer une instance d'un gestionnaire de files d'attente existant ou pour supprimer les informations de configuration d'un gestionnaire de files d'attente qui a été supprimé d'un autre serveur.



La commande **CRTMQM** comporte trois paramètres pour vous aider à configurer un gestionnaire de files d'attente multi-instance.

#### **MQMDIRP ( \*DFT | *préfixe-répertoire* )**

Utilisez ce paramètre pour sélectionner un point de montage mappé aux données du gestionnaire de files d'attente sur le stockage réseau.

#### **ASP ( \*SYSTEM|\*ASPDEV| *numéro-pool-mémoire-secondaire* )**

Indiquez \*SYSTEM ou un *numéro de pool de mémoire secondaire* pour placer le journal du gestionnaire de files d'attente sur le système ou sur un ASP utilisateur de base. Sélectionnez l'option \*ASPDEV et définissez un nom d'unité à l'aide du paramètre **ASPDEV** pour placer le journal du gestionnaire de files d'attente sur un ASP indépendant.

#### **ASPDEV ( \*ASP|*nom-unité* )**

Indiquez le *nom d'unité* d'une unité ASP indépendante principale ou secondaire. La sélection de \*ASP a le même résultat que la spécification de **ASP** (\*SYSTEM).

### **IBM i Remarques sur les performances et la reprise sur disque sous IBM i**

Utilisez différents pools de mémoire secondaire pour améliorer les performances et la fiabilité.

Si vous utilisez un grand nombre de messages persistants ou de messages volumineux dans vos applications, le temps passé à écrire ces messages sur le disque devient un facteur important dans les performances du système.

Vérifiez que vous disposez d'une activation de disque suffisante pour faire face à cette possibilité ou envisagez un pool de mémoire secondaire (ASP) distinct dans lequel placer les récepteurs de journal de votre gestionnaire de files d'attente.

Vous pouvez spécifier l'ASP dans lequel la bibliothèque de votre gestionnaire de files d'attente et les journaux sont stockés lorsque vous créez votre gestionnaire de files d'attente à l'aide du paramètre ASP de **CRTMQM**. Par défaut, la bibliothèque du gestionnaire de files d'attente, les journaux et les données IFS sont stockés dans l'ASP système.

Les ASP permettent d'isoler des objets sur une ou plusieurs unités de disque spécifiques. Cela peut également réduire la perte de données en raison d'une défaillance de support de disque. Dans la plupart des cas, seules les données stockées sur les unités de disque de l'ASP concerné sont perdues.

Il est recommandé de stocker la bibliothèque du gestionnaire de files d'attente et les données de journal dans des ASP utilisateur distincts du système de fichiers IFS racine afin de fournir une reprise en ligne et de réduire les conflits d'accès au disque.

Pour plus d'informations, voir [Backup and recovery](#) dans la documentation IBM i.

### **IBM i Utilisation de SAVLIB pour sauvegarder les bibliothèques IBM MQ sous IBM i**

Vous ne pouvez pas utiliser SAVLIB LIB(\*ALLUSR) pour sauvegarder les bibliothèques IBM MQ, car ces bibliothèques ont des noms commençant par Q.

Vous pouvez utiliser SAVLIB LIB(QM\*) pour sauvegarder toutes les bibliothèques du gestionnaire de files d'attente, mais uniquement si vous utilisez une unité de sauvegarde autre que \*SAVF. Pour DEV(\*SAVF), vous devez utiliser une commande SAVLIB pour chaque bibliothèque de gestionnaire de files d'attente de votre système.

### **IBM i mise au repos IBM MQ for IBM i**

Cette section explique comment mettre au repos (arrêt approprié) IBM MQ for IBM i.

Pour mettre au repos IBM MQ for IBM i:

1. Connectez-vous à une nouvelle session IBM MQ for IBM i interactive en vous assurant que vous n'accédez à aucun objet.

2. Assurez-vous que vous disposez :

- Droit \*ALLOBJ ou droit de gestion d'objet pour la bibliothèque QMQM
- Droits suffisants pour utiliser la commande ENDSBS

3. Informez tous les utilisateurs que vous allez arrêter IBM MQ for IBM i.

4. La procédure à suivre varie selon que vous souhaitez arrêter (mettre au repos) un seul gestionnaire de files d'attente (où d'autres gestionnaires de files d'attente peuvent exister) (voir «[Arrêt d'un gestionnaire de files d'attente unique pour IBM MQ for IBM i](#)», à la page 466 ). ou tous les gestionnaires de files d'attente (voir «[Arrêt de tous les gestionnaires de files d'attente pour IBM MQ for IBM i](#)», à la page 468 ).

5. Arrêtez le serveur mqweb en entrant la commande suivante dans qshell:

```
/QIBM/ProdData/mqm/bin/endmqweb
```

## ENDMQM, paramètre ENDCCTJOB (\*YES)

Le paramètre ENDMQM ENDCCTJOB (\*YES) fonctionne différemment dans IBM MQ for IBM i V6.0 et ultérieure par rapport aux versions précédentes.

Dans les versions précédentes, lorsque vous spécifiez ENDCCTJOB (\*YES), MQ force l'arrêt de vos applications pour vous.

Sous IBM MQ for IBM i V6.0 ou ultérieure, lorsque vous indiquez ENDCCTJOB (\*YES), vos applications ne sont pas arrêtées mais sont déconnectées du gestionnaire de files d'attente.

Si vous spécifiez ENDCCTJOB (\*YES) et que vous avez des applications qui ne sont pas écrites pour détecter qu'un gestionnaire de files d'attente est en cours d'arrêt, lors de la prochaine émission d'un nouvel appel MQI, l'appel sera renvoyé avec une erreur MQRC\_CONNECTION\_BROKEN (2009).

Au lieu d'utiliser ENDCCTJOB (\*YES), utilisez le paramètre ENDCCTJOB (\*NO) et l'option 22 de la commande WRKMQM (Gestion des travaux) pour arrêter manuellement les travaux d'application qui empêchent le redémarrage d'un gestionnaire de files d'attente.

## Arrêt d'un gestionnaire de files d'attente unique pour IBM MQ for IBM i

Utilisez ces informations pour comprendre les trois types d'arrêt.

Dans les procédures qui suivent, nous utilisons un exemple de nom de gestionnaire de files d'attente QMgr1 et un exemple de nom de sous-système SUBX. Si nécessaire, remplacez ces noms par vos propres valeurs.

### Arrêt planifié

Arrêt planifié d'un gestionnaire de files d'attente sous IBM i

1. Avant l'arrêt, exécutez:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNDTA(*YES)
```

2. Pour arrêter le gestionnaire de files d'attente, exécutez:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Si QMgr1 ne s'arrête pas, le canal ou les applications sont probablement occupés.

3. Si vous devez arrêter QMgr1 immédiatement, exécutez la commande suivante:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

## Arrêt non planifié

1. Pour arrêter le gestionnaire de files d'attente, exécutez:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Si QMgr1 ne s'arrête pas, le canal ou les applications sont probablement occupés.

2. Si vous devez arrêter QMgr1 immédiatement, exécutez ce qui suit:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(15)
```

## Arrêt en cas de conditions anormales

1. Pour arrêter le gestionnaire de files d'attente, exécutez:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Si QMgr1 ne s'arrête pas, passez à l'étape 3 en indiquant que:

- QMgr1 se trouve dans son propre sous-système, ou
  - Vous pouvez arrêter tous les gestionnaires de files d'attente qui partagent le même sous-système que QMgr1. Utilisez la procédure d'arrêt non planifié pour tous les gestionnaires de files d'attente de ce type.
2. Une fois que vous avez effectué toutes les étapes de la procédure pour tous les gestionnaires de files d'attente partageant le sous-système ( SUBX dans nos exemples), exécutez:

```
ENDSBS SUBX *IMMED
```

Si cette commande échoue, arrêtez tous les gestionnaires de files d'attente à l'aide de la procédure d'arrêt non planifiée et effectuez un IPL sur votre machine.

**Avertissement :** N'utilisez pas les travaux ENDJOBABN for IBM MQ qui ne parviennent pas à s'arrêter suite à ENDJOB ou ENDSBS, sauf si vous êtes prêt à effectuer un IPL sur votre machine immédiatement après.

3. Démarrez le sous-système en exécutant:

```
STRSBS SUBX
```

4. Arrêtez immédiatement le gestionnaire de files d'attente en exécutant:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)  
ENDCCTJOB(*YES) TIMEOUT(10)
```

5. Redémarrez le gestionnaire de files d'attente en exécutant:

```
STRMQM MQMNAME(QMgr1)
```

Si cela échoue, et que vous:

- Vous avez redémarré votre machine en exécutant un IPL, ou
- N'avoir qu'un seul gestionnaire de files d'attente

Rangez la mémoire partagée IBM MQ en exécutant:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

avant de répéter l'étape 5.

Si le redémarrage du gestionnaire de files d'attente prend plus de quelques secondes, IBM MQ ajoute des messages d'état par intermittence au journal des travaux détaillant la progression du démarrage.

Si vous rencontrez toujours des problèmes lors du redémarrage de votre gestionnaire de files d'attente, contactez le support IBM . Toute autre action que vous pourriez effectuer pourrait endommager le gestionnaire de files d'attente, ce qui ne permet pas à IBM MQ d'effectuer de reprise.

## Arrêt de tous les gestionnaires de files d'attente pour IBM MQ for IBM i

Utilisez ces informations pour comprendre les trois types d'arrêt.

Les procédures sont presque les mêmes que pour un gestionnaire de files d'attente unique, mais en utilisant \*ALL à la place du nom du gestionnaire de files d'attente lorsque cela est possible, et en utilisant une commande à plusieurs reprises en utilisant chaque nom de gestionnaire de files d'attente à tour de rôle. Tout au long des procédures, nous utilisons un exemple de nom de gestionnaire de files d'attente QMgr1 et un exemple de nom de sous-système SUBX. Remplacez-les par les vôtres.

### Arrêt planifié

1. Une heure avant l'arrêt, exécutez:

```
RCDMQMIMG OBJ(*ALL) OBJTYPE(*ALL) MQMNAME(QMgr1) DSPJRNTA(*YES)
```

Répétez cette opération pour chaque gestionnaire de files d'attente que vous souhaitez arrêter.

2. Pour arrêter le gestionnaire de files d'attente, exécutez:

```
ENDMQM MQMNAME(QMgr1) OPTION(*CNTRLD)
```

Répétez cette opération pour chaque gestionnaire de files d'attente que vous souhaitez arrêter ; des commandes distinctes peuvent être exécutées en parallèle.

Si un gestionnaire de files d'attente ne s'arrête pas dans un délai raisonnable (par exemple, 10 minutes), passez à l'étape 3.

3. Pour arrêter immédiatement tous les gestionnaires de files d'attente, exécutez la commande suivante:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

### Arrêt non planifié

1. Pour arrêter un gestionnaire de files d'attente, exécutez:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Répétez cette opération pour chaque gestionnaire de files d'attente que vous souhaitez arrêter ; des commandes distinctes peuvent être exécutées en parallèle.

Si les gestionnaires de files d'attente ne s'arrêtent pas, le canal ou les applications sont probablement occupés.

2. Si vous devez arrêter les gestionnaires de files d'attente immédiatement, exécutez la commande suivante:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

## Arrêt en cas de conditions anormales

1. Pour arrêter les gestionnaires de files d'attente, exécutez:

```
ENDMQM MQMNAME(QMgr1) OPTION(*IMMED)
```

Répétez cette opération pour chaque gestionnaire de files d'attente que vous souhaitez arrêter ; des commandes distinctes peuvent être exécutées en parallèle.

2. Arrêtez les sous-systèmes ( SUBX dans nos exemples) en exécutant:

```
ENDSBS SUBX *IMMED
```

Répétez cette opération pour chaque sous-système que vous souhaitez arrêter ; des commandes distinctes peuvent être exécutées en parallèle.

Si cette commande échoue, effectuez un IPL sur votre système.

**Avertissement :** N'utilisez pas ENDJOBABN pour les travaux qui ne parviennent pas à s'arrêter suite à ENDJOB ou ENDSBS, sauf si vous êtes prêt à effectuer un IPL sur votre système immédiatement après.

3. Démarrez les sous-systèmes en exécutant:

```
STRSBS SUBX
```

Répétez cette opération pour chaque sous-système que vous souhaitez démarrer.

4. Arrêtez immédiatement les gestionnaires de files d'attente en exécutant:

```
ENDMQM MQMNAME(*ALL) OPTION(*IMMED)
ENDCCTJOB(*YES) TIMEOUT(15)
```

5. Redémarrez les gestionnaires de files d'attente en exécutant:

```
STRMQM MQMNAME(QMgr1)
```

Répétez cette opération pour chaque gestionnaire de files d'attente que vous souhaitez démarrer.

Si le redémarrage d'un gestionnaire de files d'attente prend plus de quelques secondes, IBM MQ affiche des messages d'état détaillant par intermittence la progression du démarrage.

Si vous rencontrez toujours des problèmes lors du redémarrage d'un gestionnaire de files d'attente, contactez le support IBM . Toute autre action que vous pourriez effectuer pourrait endommager les gestionnaires de files d'attente, laissant MQSeries ou IBM MQ dans l'impossibilité d'effectuer une reprise.

## Administering IBM MQ for z/OS

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

For details of how to administer IBM MQ for z/OS and the different administrative tasks you might have to undertake, see the following links.

You can also administer IBM MQ for z/OS using the IBM MQ Explorer running in a Linux shell. For more information, see [“Administration à l'aide de IBM MQ Explorer”](#) on page 125.

### **Related concepts**

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

### **Related tasks**

[“Administration de IBM MQ”](#) on page 7

Pour administrer vos gestionnaires de files d'attente IBM MQ et les ressources associées, choisissez la méthode de votre choix parmi un ensemble de tâches que vous pouvez utiliser pour activer et gérer ces ressources.

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

## **z/OS Issuing queue manager commands on z/OS**

You can control most of the operational environment of IBM MQ by using control commands. You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or authorized applications.

### **About this task**

You use MQSC commands, in batch or interactive mode, to administer queue managers directly. You use PCF commands to help you create applications that administer queue managers. MQSC commands are in human-readable text form, whereas PCF commands let applications create requests and read the replies without having to parse text strings. Like MQSC commands, applications issue PCF commands by sending them as messages to the command input queue.

The following topics describe how you issue queue manager commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications.

Not all commands can be issued from all sources. See [“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS”](#) on page 470.

### **Related tasks**

[Preparing sample applications for the TSO environment on z/OS](#)

### **Related information**

[Administering IBM MQ using MQSC commands](#)

## **z/OS Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS**

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

### **Which MQSC and PCF commands can control each IBM MQ object**

Table 1 in [“Command summary for IBM MQ for z/OS”](#) maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object. See also [MQSC commands reference](#) and [“Utilisation des formats de commande programmables IBM MQ”](#) on page 28.

### **List of sources from which commands can be issued**

If you are a suitably authorized user, you can issue IBM MQ commands from the following sources:

- The z/OS console or equivalent (such as SDSF/TSO).

See also [“Using the operations and control panels on z/OS”](#) on page 484.

**Note:** When using the z/OS console, you need to add /cpf to the start of a command, where cpf is the command prefix for the queue manager subsystem.

- The initialization input data sets CSQINP1, CSQINP2, CSQINPT and CSQINPX.

See [“Initialization commands for IBM MQ for z/OS”](#) on page 481.

- The z/OS master get command routine, MGCRE (SVC 34).
- The IBM MQ batch utility programs such as CSQUTIL, which processes a list of commands in a sequential data set.

See [“Using the IBM MQ for z/OS utilities”](#) on page 492.

- Suitably authorized applications, sending commands as messages to the SYSTEM.COMMAND.INPUT queue.

The application can be any of the following:

- A batch region program
- A CICS application
- An IMS application
- A TSO application
- An application program or utility on another IBM MQ system

See [“Writing programs to administer IBM MQ for z/OS”](#) on page 501 and [Preparing sample applications for the TSO environment on z/OS](#).

## Not all commands can be issued from all sources

Commands are classified according to where they can be issued from:

**1**

CSQINP1

**2**

CSQINP2

**C**

The z/OS console

**R**

The command server and command queue, by means of CSQUTIL, CSQINPT, CSQINPX, or authorized applications.

Within the command descriptions in [MQSC commands reference](#), these sources are identified by the use of the characters 1, 2, C, and R in each command description. Table 2 in ["Command summary for IBM MQ for z/OS"](#) summarizes the MQSC commands and the sources from which they can be issued.

### Related tasks

[Preparing sample applications for the TSO environment on z/OS](#)

### Related information

[Administering IBM MQ using MQSC commands](#)

## Command summary for IBM MQ for z/OS

A summary of the main MQSC and PCF commands, and of the sources from which you can run MQSC commands on IBM MQ for z/OS.

[Table 25 on page 472](#) maps which MQSC and PCF commands can be used on IBM MQ for z/OS to alter, define, delete and display each IBM MQ object.

Table 25. Summary of the main MQSC and PCF commands by object type

MQSC command	ALTER	DEFINE	DISPLAY	DELETE
PCF command	Change	Create/Copy	Inquire	Delete
AUTHINFO	X	X	X	X
CFSTATUS			X	
CFSTRUCT	X	X	X	X
CHANNEL	X	X	X	X
CHSTATUS			X	
NAMELIST	X	X	X	X
PROCESS	X	X	X	X
QALIAS	M	M	M	M
QCLUSTER			M	
QLOCAL	M	M	M	M
QMGR	X		X	
QMODEL	M	M	M	M
QREMOTE	M	M	M	M
QUEUE	P	P	X	P
QSTATUS			X	
STGCLASS	X	X	X	X

**Key to table symbols:**

- M = MQSC only
- P = PCF only
- X = both

There are many other MQSC and PCF commands which allow you to manage other IBM MQ resources, and carry out other actions in addition to those summarized in [Table 25 on page 472](#).

[Table 26 on page 472](#) shows every MQSC command, and where each command can be issued from.

- CSQINP1 initialization input data set
- CSQINP2 initialization input data set
- z/OS console (or equivalent)
- SYSTEM.COMMAND.INPUT queue and command server (from applications, CSQUTIL, or the CSQINPX initialization input data set)

Table 26. Sources from which to run MQSC commands

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
ALTER AUTHINFO		X	X	X
ALTER BUFFPOOL		X	X	X
ALTER CFSTRUCT		X	X	X



Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
ALTER CHANNEL		X	X	X
ALTER NAMELIST		X	X	X
ALTER PROCESS		X	X	X
ALTER PSID			X	X
ALTER QALIAS		X	X	X
ALTER QLOCAL		X	X	X
ALTER QMGR		X	X	X
ALTER QMODEL		X	X	X
ALTER QREMOTE		X	X	X
ALTER SECURITY	X	X	X	X
ALTER SMDS		X	X	X
ALTER STGCLASS		X	X	X
ALTER SUB			X	X
ALTER TOPIC		X	X	X
ALTER TRACE	X	X	X	X
ARCHIVE LOG	X	X	X	X
BACKUP CFSTRUCT			X	X
CLEAR QLOCAL		X	X	X
CLEAR TOPICSTR			X	X
DEFINE AUTHINFO		X	X	X
DEFINE BUFFPOOL	X			
DEFINE CFSTRUCT		X	X	X
DEFINE CHANNEL		X	X	X
DEFINE LOG			X	X
DEFINE MAXSMGS		X	X	X
DEFINE NAMELIST		X	X	X
DEFINE PROCESS		X	X	X
DEFINE PSID	X		X	X
DEFINE QALIAS		X	X	X
DEFINE QLOCAL		X	X	X
DEFINE QMODEL		X	X	X
DEFINE QREMOTE		X	X	X
DEFINE STGCLASS		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DEFINE SUB			X	X
DEFINE TOPIC		X	X	X
DELETE AUTHINFO		X	X	X
DELETE BUFFPOOL		X	X	X
DELETE CFSTRUCT		X	X	X
DELETE CHANNEL			X	X
DELETE NAMELIST		X	X	X
DELETE PROCESS		X	X	X
DELETE PSID			X	X
DELETE QALIAS		X	X	X
DELETE QLOCAL		X	X	X
DELETE QMODEL		X	X	X
DELETE QREMOTE		X	X	X
DELETE STGCLASS		X	X	X
DELETE SUB			X	X
DELETE TOPIC		X	X	X
DISPLAY ARCHIVE	X	X	X	X
DISPLAY AUTHINFO		X	X	X
DISPLAY CFSTATUS			X	X
DISPLAY CFSTRUCT		X	X	X
DISPLAY CHANNEL		X	X	X
DISPLAY CHINIT			X	X
DISPLAY CHLAUTH		X	X	X
DISPLAY CHSTATUS			X	X
DISPLAY CLUSQMGR			X	X
DISPLAY CMDSERV	X	X	X	X
DISPLAY CONN		X	X	X
DISPLAY GROUP		X	X	X
DISPLAY LOG	X	X	X	X
DISPLAY MAXSMSGS		X	X	X
DISPLAY NAMELIST		X	X	X
DISPLAY PROCESS		X	X	X
DISPLAY PUBSUB		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
DISPLAY QALIAS		X	X	X
DISPLAY QCLUSTER		X	X	X
DISPLAY QLOCAL		X	X	X
DISPLAY QMGR		X	X	X
DISPLAY QMODEL		X	X	X
DISPLAY QREMOTE		X	X	X
DISPLAY QSTATUS		X	X	X
DISPLAY QUEUE		X	X	X
DISPLAY SBSTATUS			X	X
DISPLAY SECURITY			X	X
DISPLAY SMDS		X	X	X
DISPLAY SMDSCONN		X	X	X
DISPLAY STGCLASS		X	X	X
DISPLAY SUB			X	X
DISPLAY SYSTEM	X	X	X	X
DISPLAY TCLUSTER		X	X	X
DISPLAY THREAD		X	X	X
DISPLAY TOPIC		X	X	X
DISPLAY TPSTATUS		X	X	X
DISPLAY TRACE	X	X	X	X
DISPLAY USAGE		X	X	X
MOVE QLOCAL		X	X	X
PING CHANNEL			X	X
RECOVER BSDS			X	X
RECOVER CFSTRUCT			X	X
REFRESH CLUSTER			X	X
REFRESH QMGR		X	X	X
REFRESH SECURITY			X	X
RESET CFSTRUCT			X	X
RESET CHANNEL			X	X
RESET CLUSTER			X	X
RESET QMGR		X	X	X
RESET QSTATS		X	X	X

Table 26. Sources from which to run MQSC commands (continued)

Command	CSQINP1	CSQINP2	z/OS console	Command input queue and server
RESET SMDS			X	X
RESET TPIPE			X	X
RESOLVE CHANNEL			X	X
RESOLVE INDOUBT		X	X	X
RESUME QMGR			X	X
RVERIFY SECURITY		X	X	X
SET ARCHIVE	X	X	X	X
SET CHLAUTH		X	X	X
SET LOG	X	X	X	X
SET SYSTEM	X	X	X	X
START CHANNEL			X	X
START CHINIT		X	X	X
START CMDSERV	X	X	X	
START LISTENER			X	X
START QMGR			X	
START SMDSCONN		X	X	X
START TRACE	X	X	X	X
STOP CHANNEL			X	X
STOP CHINIT			X	X
STOP CMDSERV	X	X	X	
STOP LISTENER			X	X
STOP QMGR			X	X
STOP SMDSCONN		X	X	X
STOP TRACE	X	X	X	X
SUSPEND QMGR			X	X

In MQSC commands, each command description identifies the sources from which that command can be run.

### Using MQSC to start and stop a queue manager on z/OS

An introduction to using control commands on IBM MQ for z/OS: After you have installed IBM MQ, use MQSC commands to start and stop a queue manager.

## Before you begin

After you have installed IBM MQ, it is defined as a formal z/OS subsystem. This message appears during any initial program load (IPL) of z/OS:

```
CSQ3110I +CSQ1 CSQ3UR00 - SUBSYSTEM ssnm INITIALIZATION COMPLETE
```

where *ssnm* is the IBM MQ subsystem name.

From now on, you can start the queue manager for that subsystem *from any z/OS console that has been authorized to issue system control commands*; that is, a z/OS SYS command group. You must issue the START command from the authorized console, you cannot issue it through JES or TSO.

If you are using queue sharing groups, you must start RRS first, and then Db2®, before you start the queue manager.

## About this task

When a queue manager stops under normal conditions, its last action is to take a termination checkpoint. This checkpoint, and the logs, give the queue manager the information it needs to restart.

The following steps contain information about the START and STOP commands, and contain a brief overview of start-up after an abnormal termination has occurred.

## Procedure

### 1. Start a queue manager

You start a queue manager by issuing a START QMGR command. However, you cannot successfully use the START command unless you have appropriate authority. See the [Setting up security on z/OS](#) for information about IBM MQ security. The following code shows examples of the START command. Note that you must prefix an MQSC command with a command prefix string (CPF).

```
+CSQ1 START QMGR
+CSQ1 START QMGR PARM(NEWLOG)
```

See [START QMGR](#) for information about the syntax of the START QMGR command.

You cannot run the queue manager as a batch job or start it using a z/OS command START. These methods are likely to start an address space for IBM MQ that then ends abnormally. Nor can you start a queue manager from the CSQUTIL utility program or a similar user application.

You can, however, start a queue manager from an APF-authorized program by passing a START QMGR command to the z/OS MGCRE (SVC 34) service.

If you are using queue sharing groups, the associated Db2 systems and RRS must be active when you start the queue manager.

### Start options

When you start a queue manager, a system parameter module is loaded. You can specify the name of the system parameter module in one of two ways:

- With the PARM parameter of the /cpf START QMGR command, for example

```
/cpf START QMGR PARM(CSQ1ZPRM)
```

- With a parameter in the startup procedure, for example, code the JCL EXEC statement as

```
//MQM EXEC PGM=CSQYASCP,PARM='ZPARAM(CSQ1ZPRM)'
```

A system parameter module provides information specified when the queue manager was customized.

You can use the **QMGRPROD** option to specify the product against which the queue manager usage is to be recorded, and the **AMSPROD** option to specify the equivalent for AMS if that is used. See the `MQSC START QMGR` command for details of the permitted values.

An example JCL EXEC statement follows:

```
//MQM EXEC PGM=CSQYASCP,PARM='QMGRPROD(MQ)'
```

See [z/OS MVS Product Management](#) for more information on measured usage and product registration.

You can also use the `ENVPARM` option to substitute one or more parameters in the JCL procedure for the queue manager.

For example, you can update your queue manager startup procedure, so that the **DDname** `CSQINP2` is a variable. This means that you can change the `CSQINP2` **DDname** without changing the startup procedure. This is useful for implementing changes, providing back-outs for operators, and queue manager operations.

Suppose your start-up procedure for queue manager `CSQ1` looked like this:

```
//CSQ1MSTR PROC INP2=NORM
//MQMESA EXEC PGM=CSQYASCP
//STEPLIB DD DISP=SHR,DSN=th1qual.SCSQANLE
// DD DISP=SHR,DSN=th1qual.SCSQAUTH
// DD DISP=SHR,DSN=db2qual.SDSNLOAD
//BDS1 DD DISP=SHR,DSN=myqual.BSDS01
//BDS2 DD DISP=SHR,DSN=myqual.BSDS02
//CSQP0000 DD DISP=SHR,DSN=myqual.PSID00
//CSQP0001 DD DISP=SHR,DSN=myqual.PSID01
//CSQP0002 DD DISP=SHR,DSN=myqual.PSID02
//CSQP0003 DD DISP=SHR,DSN=myqual.PSID03
//CSQINP1 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1INP1)
//CSQINP2 DD DISP=SHR,DSN=myqual.CSQINP(CSQ1&INP2.)
//CSQOUT1 DD SYSOUT=*
//CSQOUT2 DD SYSOUT=*
```

If you then start your queue manager with the following command:

```
+CSQ1 START QMGR
```

then the `CSQINP2` used is a member called `CSQ1NORM`.

However, suppose you are putting a new suite of programs into production so that the next time you start queue manager `CSQ1`, the `CSQINP2` definitions are to be taken from member `CSQ1NEW`. To do this, you would start the queue manager with this command:

```
+CSQ1 START QMGR ENVPARM('INP2=NEW')
```

and `CSQ1NEW` would be used instead of `CSQ1NORM`. Note: z/OS limits the `KEYWORD=value` specifications for symbolic parameters (as in `INP2=NEW`) to 255 characters.

### Starting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting a queue manager after it ends abnormally is different from starting it after the `STOP QMGR` command has been issued. After `STOP QMGR`, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

However, if the queue manager ends abnormally, it terminates without being able to finish its work or take a termination checkpoint. When you restart a queue manager after an abend, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks. Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies.

### User messages on start-up

When you start a queue manager successfully, the queue manager produces a set of startup messages.

#### 2. Stop a queue manager.

Before stopping a queue manager, all IBM MQ-related write-to-operator-with-reply (WTOR) messages must receive replies, for example, getting log requests. Each of the following commands terminates a running queue manager.

```
+CSQ1 STOP QMGR
+CSQ1 STOP QMGR MODE(QUIESCE)
+CSQ1 STOP QMGR MODE(FORCE)
+CSQ1 STOP QMGR MODE(RESTART)
```

The command STOP QMGR defaults to STOP QMGR MODE(QUIESCE).

In QUIESCE mode, IBM MQ does not allow any new connection threads to be created, but allows existing threads to continue; it terminates only when all threads have ended. Applications can request to be notified in the event of the queue manager quiescing. Therefore, use the QUIESCE mode where possible so that applications that have requested notification have the opportunity to disconnect. See [What happens during termination](#) for details.

If the queue manager does not terminate in a reasonable time in response to a STOP QMGR MODE(QUIESCE) command, use the DISPLAY CONN command to determine whether any connection threads exist, and take the necessary steps to terminate the associated applications. If there are no threads, issue a STOP QMGR MODE(FORCE) command.

The STOP QMGR MODE(QUIESCE) and STOP QMGR MODE(FORCE) commands deregister IBM MQ from the MVS Automatic Restart Manager (ARM), preventing ARM from restarting the queue manager automatically. The STOP QMGR MODE(RESTART) command works in the same way as the STOP QMGR MODE(FORCE) command, except that it does not deregister IBM MQ from ARM. This means that the queue manager is eligible for immediate automatic restart.

If the IBM MQ subsystem is not registered with ARM, the STOP QMGR MODE(RESTART) command is rejected and the following message is sent to the z/OS console:

```
CSQY205I ARM element arm-element is not registered
```

If this message is not issued, the queue manager is restarted automatically. For more information about ARM, see [“Using the z/OS Automatic Restart Manager \(ARM\)”](#) on page 559.

#### **Only cancel the queue manager address space if STOP QMGR MODE(FORCE) does not terminate the queue manager.**

If a queue manager is stopped by either canceling the address space or by using the command STOP QMGR MODE(FORCE), consistency is maintained with connected CICS or IMS systems. Resynchronization of resources is started when a queue manager restarts and is completed when the connection to the CICS or IMS system is established.

**Note:** When you stop your queue manager, you might find message IEF352I is issued. z/OS issues this message if it detects that failing to mark the address space as unusable would lead to an integrity exposure. You can ignore this message.

## Stop messages

After issuing a STOP QMGR command, you get the messages CSQY009I and CSQY002I, for example:

```
CSQY009I +CSQ1 ' STOP QMGR' COMMAND ACCEPTED FROM
USER(userid), STOP MODE(FORCE)
CSQY002I +CSQ1 QUEUE MANAGER STOPPING
```

Where `userid` is the user ID that issued the STOP QMGR command, and the MODE parameter depends on that specified in the command.

When the STOP command has completed successfully, the following messages are displayed on the z/OS console:

```
CSQ9022I +CSQ1 CSQYASCP ' STOP QMGR' NORMAL COMPLETION
CSQ3104I +CSQ1 CSQ3EC0X - TERMINATION COMPLETE
```

If you are using ARM, and you did not specify MODE(RESTART), the following message is also displayed:

```
CSQY204I +CSQ1 ARM DEREGISTER for element arm-element type
arm-element-type successful
```

You cannot restart the queue manager until the following message has been displayed:

```
CSQ3100I +CSQ1 CSQ3EC0X - SUBSYSTEM ssnm READY FOR START COMMAND
```

## Issuing commands from a z/OS console or equivalent

You can issue IBM MQ MQSC and PCF commands from a z/OS console or its equivalent. You can also issue IBM MQ commands from anywhere where you can issue z/OS commands, such as SDSF or by a program using the MGCRC macro. When using the z/OS console, you add `/cpf` to the start of a command.

### Before you begin

Not all commands can be issued by the z/OS console. Within the command description topics (the children of [MQSC commands reference](#)), each command that can be issued by the console is identified by the character 'C'. [Table 2 in "Command summary for IBM MQ for z/OS"](#) summarizes the MQSC commands and the sources from which they can be issued.

You cannot issue IBM MQ commands using the IMS/SSR command format from an IMS terminal. This function is not supported by the IMS adapter.

The input field provided by SDSF might not be long enough for some commands, particularly those commands for channels.

The maximum amount of data that can be displayed as a result of a command typed in at the console is 32 KB.

### About this task

If you are a suitably authorized user, you can issue IBM MQ commands from the z/OS console or equivalent (such as SDSF/TSO).

When using the z/OS console, you need to add `/cpf` to the start of a command, where `cpf` is the command prefix for the queue manager subsystem.

The following steps refer to commands and attributes using their MQSC command names rather than their PCF names.



## Procedure

- Use command prefix strings

Each IBM MQ command must be prefixed with a command prefix string (CPF).

Because more than one IBM MQ subsystem can run under z/OS, the CPF is used to indicate which IBM MQ subsystem processes the command.

For example, to start the queue manager for a subsystem called CSQ1, where CPF is ' +CSQ1 ', you issue the following command from the operator console:

```
+CSQ1 START QMGR
```

This CPF must be defined in the subsystem name table (for the subsystem CSQ1), as described in [Defining command prefix strings \(CPFs\)](#). In the examples, the string +CSQ1 is used as the command prefix.

- Use the z/OS console to issue commands

You can type simple commands from the z/OS console, for example:

```
+CSQ1 DISPLAY QUEUE(TRANSMIT.QUEUE.PROD) TYPE(QLOCAL)
```

However, for complex commands or for sets of commands that you issue frequently, the other methods of issuing commands are better.

- Receive command responses

Direct responses to commands are sent to the console that issued the command. IBM MQ supports the *Extended Console Support* (EMCS) function available in z/OS, and therefore consoles with 4 byte IDs can be used. Additionally, all commands except START QMGR and STOP QMGR support the use of Command and Response Tokens (CARTs) when the command is issued by a program using the MGCRC macro.

## Related tasks

[“Using the operations and control panels on z/OS” on page 484](#)

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

[Preparing sample applications for the TSO environment on z/OS](#)

## Initialization commands for IBM MQ for z/OS

Initialization commands can be used to control the queue manager startup.

Commands in the initialization input data sets are processed when IBM MQ is initialized on queue manager startup. Three types of command can be issued from the initialization input data sets:

- Commands to define IBM MQ entities that cannot be defined elsewhere, for example DEFINE BUFFPOOL.

These commands must reside in the data set identified by the DD name CSQINP1. They are processed before the restart phase of initialization. They cannot be issued through the console, operations and control panels, or an application program. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT1 statement of the started task procedure.

- Commands to define IBM MQ objects that are recoverable after restart. These definitions must be specified in the data set identified by the DD name CSQINP2. They are stored in page set zero. CSQINP2 is processed after the restart phase of initialization. The responses to these commands are written to the sequential data set that you refer to in the CSQOUT2 statement of the started task procedure.
- Commands to manipulate IBM MQ objects. These commands must also be specified in the data set identified by the DD name CSQINP2. For example, the IBM MQ-supplied sample contains an ALTER

QMGR command to specify a dead-letter queue for the subsystem. The response to these commands is written to the CSQOUT2 output data set.

**Note:** If IBM MQ objects are defined in CSQINP2, IBM MQ attempts to redefine them each time the queue manager is started. If the objects already exist, the attempt to define them fails. If you need to define your objects in CSQINP2, you can avoid this problem by using the REPLACE parameter of the DEFINE commands, however, this overrides any changes that were made during the previous run of the queue manager.

Sample initialization data set members are supplied with IBM MQ for z/OS. They are described in [Sample definitions supplied with IBM MQ](#).

## Initialization commands for distributed queuing

You can also use the CSQINP2 initialization data set for the START CHINIT command. If you need a series of other commands to define your distributed queuing environment (for example, starting listeners), IBM MQ provides a third initialization input data set, called CSQINPX, that is processed as part of the channel initiator started task procedure.

The MQSC 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. You might use the CSQINPX initialization data set to start listeners for example.

A sample channel initiator initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

## Initialization commands for publish/Subscribe

If you need a series of commands to define your publish/subscribe environment (for example, when defining subscriptions), IBM MQ provides a fourth initialization input data set, called CSQINPT.

The MQSC commands contained in the data set are executed at the end of publish/subscribe initialization, and output is written to the data set specified by the CSQOUTT DD statement. You might use the CSQINPT initialization data set to define subscriptions for example.

A sample publish/subscribe initialization data set member is supplied with IBM MQ for z/OS. It is described in [Sample definitions supplied with IBM MQ](#).

## Private and global definitions on IBM MQ for z/OS

When you define an object on IBM MQ for z/OS, you can choose whether you want to share that definition with other queue managers (a *global* definition), or whether the object definition is to be used by one queue manager only (a *private* definition). This is called the object *disposition*.

### Global definition

If your queue manager belongs to a queue sharing group, you can choose to share any object definitions you make with the other members of the group. This means that you have to define an object once only, reducing the total number of definitions required for the whole system.

Global object definitions are held in a *shared repository* (a Db2 shared database), and are available to all the queue managers in the queue sharing group. These objects have a disposition of GROUP.

### Private definition

If you want to create an object definition that is required by one queue manager only, or if your queue manager is not a member of a queue sharing group, you can create object definitions that are not shared with other members of a queue sharing group.

Private object definitions are held on page set zero of the defining queue manager. These objects have a disposition of QMGR.

You can create private definitions for all types of IBM MQ objects except CF structures (that is, channels, namelists, process definitions, queues, queue managers, storage class definitions, and authentication information objects), and global definitions for all types of objects except queue managers.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily if you want, and IBM MQ allows you to refresh the page set copies from the repository copy if required.

IBM MQ always tries to refresh the page set copies from the repository copy at startup (for channel commands, this is done when the channel initiator restarts), or if the group object is changed.

**Note:** The copy of the definition is refreshed from the definition of the group, only if the definition of the group has changed after you created the copy of the definition.

This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive. The copies are refreshed by generating DEFINE REPLACE commands, therefore there are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that changes the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that deletes that queue fails.
- If a copy of a queue would require ALTER with FORCE to change it.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers.

If the queue manager is shut down and then restarted stand-alone, any local copies of objects are deleted, unless for example, the queue has associated messages.

There is a third object disposition that applies to local queues only. This allows you to create shared queues. The definition for a shared queue is held on the shared repository and is available to all the queue managers in the queue sharing group. In addition, the messages on a shared queue are also available to all the queue managers in the queue sharing group. This is described in [Shared queues and queue sharing groups](#). Shared queues have an object disposition of SHARED.

The following table summarizes the effect of the object disposition options for queue managers started stand-alone, and as a member of a queue sharing group.

Disposition	Stand-alone queue manager	Member of a queue sharing group
QMGR	Object definition held on page set zero.	Object definition held on page set zero.
GROUP	Not allowed.	Object definition held in the shared repository. Local copy held on page set zero of each queue manager in the group.
SHARED	Not allowed.	Queue definition held in the shared repository. Messages available to any queue manager in the group.

## Manipulating global definitions

If you want to change the definition of an object that is held in the shared repository, you need to specify whether you want to change the version on the repository, or the local copy on page set zero. Use the object disposition as part of the command to do this.

### Directing commands to different queue managers on z/OS

You can use the *command scope* to control on which queue manager the command runs.

You can choose to execute a command on the queue manager where it is entered, or on a different queue manager in the queue sharing group. You can also choose to issue a particular command in parallel on all the queue managers in a queue sharing group. This is possible for both MQSC commands and PCF commands.

This is determined by the *command scope*. The command scope is used with the object disposition to determine which version of an object you want to work with.

For example, you might want to alter some of the attributes of an object, the definition of which is held in the shared repository.

- You might want to change the version on one queue manager only, and not make any changes to the version on the repository or those in use by other queue managers.
- You might want to change the version in the shared repository for future users, but leave existing copies unchanged.
- You might want to change the version in the shared repository, but also want your changes to be reflected immediately on all the queue managers in the queue sharing group that hold a copy of the object on their page set zero.

Use the command scope to specify whether the command is executed on this queue manager, another queue manager, or all queue managers. Use the object disposition to specify whether the object you are manipulating is in the shared repository (a group object), or is a local copy on page set zero (a queue manager object).

You do not have to specify the command scope and object disposition to work with a shared queue because every queue manager in the queue sharing group handles the shared queue as a single queue.

z/OS

## Using the operations and control panels on z/OS

You use these panels for defining, displaying, altering, or deleting IBM MQ objects. Use the panels for day-to-day administration and for making small changes to objects.

### Before you begin

The IBM MQ for z/OS operations and controls panels (CSQOREXX) might not support all new function and parameters added from version 7 onwards. For example, there are no panels for the direct manipulation of topic objects or subscriptions. Use one of the following supported mechanisms to administer publish/subscribe definitions and other system controls that are not directly available from other panels:

1. IBM MQ Explorer
2. z/OS console
3. Programmable Command Format (PCF) messages
4. COMMAND function of CSQUTIL
5. IBM MQ Console

Note that the generic **Command** action in the CSQOREXX panels allows you to issue any valid MQSC command, including SMDS related commands. You can use all the commands that the COMMAND function of CSQUTIL issues.

You cannot issue the IBM MQ commands directly from the command line in the panels.

To use the operations and control panels, you must have the correct security authorization; this is described in the [User IDs for command security and command resource security](#).

You cannot provide a user ID and password using CSQUTIL, or the CSQOREXX panels. Instead, if you user ID has UPDATE authority to the BATCH profile in MQCONN, you can bypass the **CHKLOCL**(REQUIRED) setting. See [Using CHKLOCL on locally bound applications](#) for more information.

If you are setting up or changing many objects, use the COMMAND function of the CSQUTIL utility program. See [“Using the CSQUTIL utility for IBM MQ for z/OS”](#) on page 494.

## About this task

The operations and control panels support the controls for the channel initiator (for example, to start a channel or a TCP/IP listener), for clustering, and for security. They also enable you to display information about threads and page set usage.

The panels work by sending MQSC type IBM MQ commands to a queue manager, through the system command input queue.

## Example

This is the panel that displays when you start a panel session:

```
IBM MQ for z/OS - Main Menu
Complete fields. Then press Enter.
Action . . . . . 1      0. List with filter  4. Manage
                          1. List or Display  5. Perform
                          2. Define like    6. Start
                          3. Alter          7. Stop
                          8. Command
Object type . . . . . CHANNEL +
Name . . . . . *
Disposition . . . . . A  Q=Qmgr, C=Copy, P=Private, G=Group,
                          S=Shared, A=All
Connect name . . . . . MQ1C - local queue manager or group
Target queue manager . . . MQ1C
                          - connected or remote queue manager for command input
Action queue manager . . . MQ1C - command scope in group
Response wait time . . . . 30  5 - 999 seconds
(C) Copyright IBM Corporation 1993, 2024. All rights reserved.
Command ==>
F1=Help      F2=Split      F3=Exit      F4=Prompt    F9=SwapNext F10=Messages
F12=Cancel
```

From this panel you can perform actions such as these:

- Choose the local queue manager you want and whether you want the commands issued on that queue manager, on a remote queue manager, or on another queue manager in the same queue sharing group as the local queue manager. Over type the queue manager name if you need to change it.
- 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 function key F1 for help about the object types if you are not sure what they are.
- Specify the disposition of the object type that you want to work with.
- 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 the action queue manager. You can then select one or more objects to work with in sequence. All the actions are available from the list.

**Note:** You are recommended to make choices that result in a list of objects being displayed, and then work from that list. Use the **Display** action, because that is allowed for all object types.

## Invocation and rules for the operations and control panels

You can control IBM MQ and issue control commands through the ISPF panels.

### How to access the IBM MQ operations and control panels

If the ISPF/PDF primary options menu has been updated for IBM MQ, you can access the IBM MQ operations and control panels from that menu. For details about updating the menu, see the [Task 20: Set up the operations and control panels](#).

You can access the IBM MQ operations and control panels from the TSO command processor panel (typically option 6 on the ISPF/PDF primary options menu). The name of the exec that you run to do this is CSQOREXX. It has two parameters; `th1qual` is the high-level qualifier for the IBM MQ libraries to be used, and `langletter` is the letter identifying the national language libraries to be used (for example, E for U.S. English). The parameters can be omitted if the IBM MQ libraries are permanently installed in your ISPF setup. Alternatively, you can issue CSQOREXX from the TSO command line.

These panels are designed to be used by operators and administrators with a minimum of formal training. Read these instructions with the panels running and try out the different tasks suggested.

**Note:** While using the panels, temporary dynamic queues with names of the form SYSTEM.CSQOREXX.\* are created.

## Rules for the operations and control panels

See [Rules for naming IBM MQ objects](#) about the general rules for IBM MQ character strings and names. However, there are some rules that apply only to the operations and control panels:

- Do not enclose strings, for example descriptions, in single or double quotation marks.
- If you include an apostrophe or quotation mark in a text field, you do not have to repeat it or add an escape character. The characters are saved exactly as you type them; for example:

This is Maria's queue

The panel processor doubles them for you to pass them to IBM MQ. However, if it has to truncate your data to do this, it does so.

- You can use uppercase or lowercase characters in most fields, and they are folded to uppercase characters when you press Enter. The exceptions are:
  - Storage class names and coupling facility structure names, which must start with uppercase A through Z and be followed by uppercase A through Z or numeric characters.
  - Certain fields that are not translated. These include:
    - Application ID
    - Description
    - Environment data
    - Object names (but if you use a lowercase object name, you might not be able to enter it at a z/OS console)
    - Remote system name
    - Trigger data
    - User data
- In names, leading blanks and leading underscores are ignored. Therefore, you cannot have object names beginning with blanks or underscores.
- Underscores are used to show the extent of blank fields. When you press Enter, trailing underscores are replaced by blanks.
- Many description and text fields are presented in multiple parts, each part being handled by IBM MQ independently. This means that trailing blanks are retained and the text is not contiguous.

### Blank fields

When you specify the **Define** action for an IBM MQ object, each field on the define panel contains a value. See the general help (extended help) for the display panels for information about where IBM MQ gets the values. If you type over a field with blanks, and blanks are not allowed, IBM MQ puts the installation default value in the field or prompts you to enter the required value.

When you specify the **Alter** action for an IBM MQ object, each field on the alter panel contains the current value for that field. If you type over a field with blanks, and blanks are not allowed, the value of that field is unchanged.

## **Objects and actions on z/OS**

The operations and control panels offer you many different types of object and a number of actions that you can perform on them.

The actions are listed on the initial panel and enable you to manipulate the objects and display information about them. These objects include all the IBM MQ objects, together with some extra ones. The objects fall into the following categories.

- [Queues, processes, authentication information objects, namelists, storage classes and CF structures](#)
- [Channels](#)
- [Cluster objects](#)
- [Queue manager and security](#)
- [Connections](#)
- [System](#)

Refer to [Actions](#) for a cross-reference table of the actions which can be taken with the IBM MQ objects.

### **Queues, processes, authentication information objects, namelists, storage classes and CF structures**

These are the basic IBM MQ objects. There can be many of each type. They can be listed, listed with filter, defined, and deleted, and have attributes that can be displayed and altered, using the LIST or DISPLAY, LIST with FILTER, DEFINE LIKE, MANAGE, and ALTER actions. (Objects are deleted using the MANAGE action.)

This category consists of the following objects:

QLOCAL	Local queue
QREMOTE	Remote queue
QALIAS	Alias queue for indirect reference to a queue
QMODEL	Model queue for defining queues dynamically
QUEUE	Any type of queue
QSTATUS	Status of a local queue
PROCESS	Information about an application to be started when a trigger event occurs
AUTHINFO	Authentication information: definitions required to perform Certificate Revocation List (CRL) checking using LDAP servers
NAMELIST	List of names, such as queues or clusters
STGCLASS	Storage class
CFSTRUCT	coupling facility (CF) structure
CFSTATUS	Status of a CF structure

### **Channels**

Channels are used for distributed queuing. There can be many of each type, and they can be listed, listed with filter, defined, deleted, displayed, and altered. They also have other functions available using the START, STOP and PERFORM actions. PERFORM provides reset, ping, and resolve channel functions.

This category consists of the following objects:

CHANNEL	Any type of channel
SENDER	Sender channel

SERVER	Server channel
RECEIVER	Receiver channel
REQUESTER	Requester channel
CLUSRCVR	Cluster-receiver channel
CLUSSDR	Cluster-sender channel
SVRCONN	Server-connection channel
CLNTCONN	Client-connection channel
CHSTATUS	Status of a channel connection

### Cluster objects

Cluster objects are created automatically for queues and channels that belong to a cluster. The base queue and channel definitions can be on another queue manager. There can be many of each type, and names can be duplicated. They can be listed, listed with filter, and displayed. PERFORM, START, and STOP are also available through the LIST actions.

This category consists of the following objects:

CLUSQ	Cluster queue, created for a queue that belongs to a cluster
CLUSCHL	Cluster channel, created for a channel that belongs to a cluster
CLUSQMGR	Cluster queue manager, the same as a cluster channel but identified by its queue manager name

Cluster channels and cluster queue managers do have the PERFORM, START and STOP actions, but only indirectly through the DISPLAY action.

### Queue manager and security

Queue manager and security objects have a single instance. They can be listed, and have attributes that can be displayed and altered (using the LIST or DISPLAY, and ALTER actions), and have other functions available using the PERFORM action.

This category consists of the following objects:

MANAGER	Queue manager: the PERFORM action provides suspend and resume cluster functions
SECURITY	Security functions: the PERFORM action provides refresh and reverify functions

### Connection

Connections can be listed, listed with filter and displayed.

This category consists only of the connection object, CONNECT.

### System

A collection of other functions. This category consists of the following objects:

SYSTEM	System functions
CONTROL	Synonym for SYSTEM

The functions available are:

LIST or DISPLAY	Display queue sharing group, distributed queuing, page set, or data set usage information.
PERFORM	Refresh or reset clustering



START Start the channel initiator or listeners  
 STOP Stop the channel initiator or listeners

## Actions

The actions that you can perform for each type of object are shown in the following table:

*Table 27. Valid operations and control panel actions for IBM MQ objects*

Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
AUTHINFO	X	X	X	X	X			
CFSTATUS				X				
CFSTRUCT	X	X	X	X	X			
CHANNEL	X	X	X	X	X	X	X	X
CHSTATUS				X	X			
CLNTCONN	X	X	X	X	X			
CLUSCHL				X	X	X(2)	X(2)	X(2)
CLUSQ				X	X			
CLUSQMGR				X	X	X(2)	X(2)	X(2)
CLUSRCVR	X	X	X	X	X	X	X	X
CLUSDR	X	X	X	X	X	X	X	X
CONNECT				X	X			
CONTROL				X		X	X	X
MANAGER	X			X		X		
NAMELIST	X	X	X	X	X			
PROCESS	X	X	X	X	X			
QALIAS	X	X	X	X	X			
QLOCAL	X	X	X	X	X			
QMODEL	X	X	X	X	X			
QREMOTE	X	X	X	X	X			
QSTATUS				X	X			
QUEUE	X	X	X	X	X			
RECEIVER	X	X	X	X	X	X	X	X
REQUESTER	X	X	X	X	X	X	X	X
SECURITY	X			X		X		
SENDER	X	X	X	X	X	X	X	X
SERVER	X	X	X	X	X	X	X	X
SVRCONN	X	X	X	X	X		X	X
STGCLASS	X	X	X	X	X			

Table 27. Valid operations and control panel actions for IBM MQ objects (continued)

Object	Alter	Define like	Manage (1)	List or Display	List with Filter	Perform	Start	Stop
SYSTEM				X		X	X	X

**Note:**

1. Provides Delete and other functions
2. Using the List or Display action

**z/OS Object dispositions on z/OS**

You can specify the *disposition* of the object with which you need to work. The disposition signifies where the object **definition** is kept, and how the object behaves.

The disposition is significant only if you are working with any of the following object types:

- queues
- channels
- processes
- namelists
- storage classes
- authentication information objects

If you are working with other object types, the disposition is disregarded.

Permitted values are:

**Q**

QMGR. The object definitions are on the page set of the queue manager and are accessible only by the queue manager.

**C**

COPY. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. They are local copies of objects defined as having a disposition of GROUP.

**P**

PRIVATE. The object definitions are on the page set of the queue manager and are accessible only by the queue manager. The objects have been defined as having a disposition of QMGR or COPY.

**G**

GROUP. The object definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

**S**

SHARED. This disposition applies only to local queues. The queue definitions are in the shared repository, and are accessible by all queue managers in the queue sharing group.

**A**

ALL. If the action queue manager is either the target queue manager, or \*, objects of **all** dispositions are included; otherwise, objects of QMGR and COPY dispositions only are included. This is the default.

**z/OS Selecting a queue manager, defaults, and levels using the ISPF control panel on z/OS**

You can use the CSQOREXX exec in ISPF to control your queue managers.

While you are viewing the initial panel, you are not connected to any queue manager. However, as soon as you press Enter, you are connected to the queue manager, or a queue manager in the queue sharing group named in the **Connect name** field. You can leave this field blank; this means that you are using the default

queue manager for batch applications. This is defined in CSQBDEFV (see [Task 19: Set up Batch, TSO, and RRS adapters](#) for information about this).

Use the **Target queue manager** field to specify the queue manager where the actions you request are to be performed. If you leave this field blank, it defaults to the queue manager specified in the **Connect name** field. You can specify a target queue manager that is not the one you connect to. In this case, you would normally specify the name of a remote queue manager object that provides a queue manager alias definition (the name is used as the *ObjectQMgrName* when opening the command input queue). To do this, you must have suitable queues and channels set up to access the remote queue manager.

The **Action queue manager** field allows you to specify a queue manager that is in the same queue sharing group as the queue manager specified in the **Target queue manager** field to be the queue manager where the actions you request are to be performed. If you specify \* in this field, the actions you request are performed on all queue managers in the queue sharing group. If you leave this field blank, it defaults to the value specified in the **Target queue manager** field. The **Action queue manager** field corresponds to using the CMDSCOPE command modifier described in [The MQSC commands](#).

### Queue manager defaults

If you leave any queue manager fields blank, or choose to connect to a queue sharing group, a secondary window opens when you press **Enter**. This window confirms the names of the queue managers you will be using. Press **Enter** to continue. When you return to the initial panel after having made some requests, you find fields completed with the actual names.

### Queue manager levels

If the action queue manager is not at IBM MQ 8.0.0 or later, some fields are not displayed, and some values cannot be entered. A few objects and actions are disallowed. In such cases, a secondary window opens asking for you to confirm that you want to proceed.

## Using the function keys and command line with the ISPF control panels on z/OS

To use the panels, you must use the function keys or enter the equivalent commands in the ISPF control panel command area.

- [Function keys](#)
  - [Processing your actions](#)
  - [“Displaying IBM MQ user messages” on page 492](#)
  - [Canceling your actions](#)
  - [Getting help](#)
- [Using the command line](#)

### Function keys

The function keys have special settings for IBM MQ. (This means that you cannot use the ISPF default values for the function keys; if you have previously used the KEYLIST OFF ISPF command anywhere, you must type KEYLIST ON in the command area of any operations and control panel and then press Enter to enable the IBM MQ settings.)

These function key settings can be displayed on the panels, as shown in [“Using the operations and control panels on z/OS” on page 484](#). If the settings are not shown, type PFSHOW in the command area of any operations and control panel and then press **Enter**. To remove the display of the settings, use the command PFSHOW OFF.

The function key settings in the operations and control panels conform to CUA standards. Although you can change the key setting through normal ISPF procedures (such as the **KEYLIST** utility), you are not recommended to do so.

**Note:** Using the **PFSHOW** and **KEYLIST** commands affects any other logical ISPF screens that you have, and their settings remain when you leave the operations and control panels.

### Processing your actions

Press **Enter** to carry out the action requested on a panel. The information from the panel is sent to the queue manager for processing.

Each time you press **Enter** in the panels, IBM MQ generates one or more operator messages. If the operation was successful, you get confirmation message CSQ9022I, otherwise you get some error messages.

### Displaying IBM MQ user messages

Press function key F10 in any panel to see the IBM MQ user messages.

### Canceling your actions

On the initial panel, both F3 and F12 exit the operations and control panels and return you to ISPF. No information is sent to the queue manager.

On any other panel, press function keys F3 or F12 to leave the current panel **ignoring any data you have typed since last pressing Enter**. Again, no information is sent to the queue manager.

- F3 takes you straight back to the initial panel.
- F12 takes you back to the previous panel.

### Getting help

Each panel has help panels associated with it. The help panels use the ISPF protocols:

- Press function key F1 on any panel to see general help (extended help) about the task.
- Press function key F1 with the cursor on any field to see specific help about that field.
- Press function key F5 from any field help panel to get the general help.
- Press function key F3 to return to the base panel, that is, the panel from which you pressed function key F1.
- Press function key F6 from any help panel to get help about the function keys.

If the help information carries on into a second or subsequent pages, a **More** indicator is displayed in the upper-right of the panel. Use these function keys to navigate through the help pages:

- F11 to get to the next help page (if there is one).
- F10 to get back to the previous help page (if there is one).

### Using the command line

You never need to use the command line to issue the commands used by the operations and control panels because they are available from function keys. The command line is provided to allow you to enter normal ISPF commands (like **PFSHOW**).

The ISPF command `PANELID ON` displays the name of the current CSQOREXX panel.

The command line is initially displayed in the default position at the bottom of the panels, regardless of what ISPF settings you have. You can use the `SETTINGS ISPF` command from any of the operations and control panels to change the position of the command line. The settings are remembered for subsequent sessions with the operations and control panels.

## Using the IBM MQ for z/OS utilities

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

IBM MQ for z/OS supplies a set of utility programs to help you perform various administrative tasks, including the following:

- Manage message security policies.
- Perform backup, restoration, and reorganization tasks.
- Issue commands and process object definitions.
- Generate data-conversion exits.
- Modify the bootstrap data set.
- List information about the logs.
- Print the logs.
- Set up Db2 tables and other Db2 utilities.
- Process messages on the dead-letter queue.

### **The message security policy utility**

The message security policy utility (CSQOUTIL) runs as a stand-alone utility to manage message security policies. See [The message security policy utility \(CSQOUTIL\)](#) for more information.

### **The CSQUTIL utility**

This is a utility program provided to help you with backup, restore and reorganize tasks. See [“Using the CSQUTIL utility for IBM MQ for z/OS”](#) on page 494.

### **The data conversion exit utility**

The IBM MQ for z/OS data conversion exit utility ( **CSQUCVX** ) runs as a stand-alone utility to create data conversion exit routines.

### **The change log inventory utility**

The IBM MQ for z/OS change log inventory utility program ( **CSQJU003** ) runs as a stand-alone utility to change the bootstrap data set (BSDS). You can use the utility to perform the following functions:

- Add or delete active or archive log data sets.
- Supply passwords for archive logs.

### **The print log map utility**

The IBM MQ for z/OS print log map utility program ( **CSQJU004** ) runs as a stand-alone utility to list the following information:

- Log data set name and log RBA association for both copies of all active and archive log data sets. If dual logging is not active, there is only one copy of the data sets.
- Active log data sets available for new log data.
- Contents of the queue of checkpoint records in the bootstrap data set (BSDS).
- Contents of the archive log command history record.
- System and utility time stamps.

### **The log print utility**

The log print utility program ( **CSQ1LOGP** ) is run as a stand-alone utility. You can run the utility specifying:

- A bootstrap data set (BSDS)

- Active logs (with no BSDS)
- Archive logs (with no BSDS)

## The queue sharing group utility

The queue sharing group utility program ( **CSQ5PQSG** ) runs as a stand-alone utility to set up Db2 tables and perform other Db2 tasks required for queue sharing groups.

## The active log preformat utility

The active log preformat utility ( **CSQJUFMT** ) formats active log data sets before they are used by a queue manager. If the active log data sets are preformatted by the utility, log write performance is improved on the queue manager's first pass through the active logs.

## The dead-letter queue handler utility

The dead-letter queue handler utility program ( **CSQUDLQH** ) runs as a stand-alone utility. It checks messages that are on the dead-letter queue and processes them according to a set of rules that you supply to the utility.

## The queue load and unload utility

The queue load and unload utility copies or moves the contents of a queue, or its messages, to a file. The utility was originally shipped as the **QLOAD** utility in IBM MQ Supportpac MO03. From IBM MQ 8.0 it is integrated into the product as executable module **CSQUDMSG** in the SCSQLOAD library, with an alias of **QLOAD** for compatibility. Sample JCL is provided as member **CSQ4QLOD** in **SCSQPROC**.

The equivalent utility for Multiplatforms is called **dmpmqmsg**. For details of the available options, including the differences for z/OS, see [dmpmqmsg \(queue load and unload\)](#).

You can also reload messages as described in [Restoring messages from a data set to a queue \(LOAD\) on z/OS](#) and [Restoring messages from a data set to a queue \(SLOAD\) on z/OS](#).

## Using the CSQUTIL utility for IBM MQ for z/OS

The CSQUTIL utility program is provided with IBM MQ for z/OS to help you perform backup, restoration, and reorganization tasks, and to issue commands and process object definitions.

### About this task

Use this utility program to invoke the following functions. For example, you can issue commands from a sequential data set using the **COMMAND** function of the CSQUTIL utility. This function transfers the commands, as messages, to the *system-command input queue* and waits for the response, which is printed together with the original commands in **SYSPRINT**.

For more information about the CSQUTIL utility program, see [IBM MQ utility program \(CSQUTIL\)](#).

### Procedure

- [COMMAND](#)

Use this function to issue MQSC commands, to record object definitions, and to make client-channel definition files.

- [COPY](#)

Use this function to read the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, and put them into a sequential file and retain the original queue.

- COPYPAGE  
Use this function to copy whole page sets to larger page sets.
- EMPTY  
Use this function to delete the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set, retaining the definitions of the queues.
- FORMAT  
Use this function to format IBM MQ for z/OS page sets.
- Restoring messages from a data set to a queue (LOAD) on z/OS  
Use this function to restore the contents of a named IBM MQ for z/OS message queue or the contents of all the queues of a named page set from a sequential file created by the COPY function.
- PAGEINFO  
Use this function to extract page set information from one or more page sets.
- RESETPAGE  
Use this function to copy whole page sets to other page set data sets and reset the log information in the copy.
- SCOPY  
Use this function to copy the contents of a queue to a data set while the queue manager is offline.
- SDEFS  
Use this function to produce a set of define commands for objects while the queue manager is offline.
- SLOAD  
Use this function to restore messages from the destination data set of an earlier COPY or SCOPY operation. SLOAD processes a single queue.
- SWITCH  
Use this function to switch or query the transmission queue associated with cluster-sender channels.

## Using the Command Facility on z/OS

Use the editor to enter or amend MQSC commands to be passed to the queue manager.

From the primary panel, CSQOPRIA, select option **8 Command**, to start the Command Facility.

You are presented with an edit session of a sequential file, *prefix.CSQUTIL.COMMANDS*, used as input to the CSQUTIL COMMAND function; see [Issuing commands to IBM MQ](#).

You do not need to prefix commands with the command prefix string (CPF).

You can continue MQSC commands on subsequent lines by terminating the current line with the continuation characters + or -. Alternatively, use line edit mode to provide long MQSC commands or the values of long attribute values within the command.

### line edit

To use line edit, move the cursor to the appropriate line in the edit panel and use **F4** to display a single line in a scrollable panel. A single line can be up to 32 760 bytes of data.

To leave line edit:

- **F3 exit** saves changes made to the line and exits
- **F12 cancel** returns to the edit panel discarding changes made to the line.

To discard changes made in the edit session, use **F12 cancel** to terminate the edit session leaving the contents of the file unchanged. Commands are not executed.

## Executing commands

When you have finished entering MQSC commands, terminate the edit session with **F3 exit** to save the contents of the file and invoke CSQUTIL to pass the commands to the queue manager. The output from command processing is held in file *prefix.CSQUTIL.OUTPUT*. An edit session opens automatically on this file so that you can view the responses. Press **F3 exit** to exit this session and return to the main menu.

## z/OS Working with IBM MQ objects on z/OS

Many of the tasks described in this documentation involve manipulating IBM MQ objects. The object types are queue managers, queues, process definitions, namelists, channels, client connection channels, listeners, services, and authentication information objects.

- [Defining simple queue objects](#)
- [Defining other types of objects](#)
- [Working with object definitions](#)
- [Working with namelists](#)

### Defining simple queue objects

To define a new object, use an existing definition as the basis for it. You can do this in one of three ways:

- By selecting an object that is a member of a list displayed as a result of options selected on the initial panel. You then enter action type 2 ( **Define like** ) in the action field next to the selected object. Your new object has the attributes of the selected object, except the disposition. You can then change any attributes in your new object as you require.
- On the initial panel, select the **Define like** action type, enter the type of object that you are defining in the **Object type** field, and enter the name of a specific existing object in the **Name** field. Your new object has the same attributes as the object you named in the **Name** field, except the disposition. You can then change any attributes in your new object definition as you require.
- By selecting the **Define like** action type, specifying an object type and then leaving the **Name** field blank. You can then define your new object and it has the default attributes defined for your installation. You can then change any attributes in your new object definition as you require.

**Note:** You do not enter the name of the object you are defining on the initial panel, but on the **Define** panel you are presented with.

The following example demonstrates how to define a local queue using an existing queue as a template.

### Defining a local queue

To define a local queue object from the operations and control panels, use an existing queue definition as the basis for your new definition. There are several panels to complete. When you have completed all the panels and you are satisfied that the attributes are correct, press Enter to send your definition to the queue manager, which then creates the actual queue.

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

<b>Action</b>	2 (Define like)
<b>Object type</b>	QLOCAL
<b>Name</b>	QUEUE.YOU.LIKE. This is the name of the queue that provides the attributes for your new queue.



Press Enter to display the **Define a Local Queue** panel. The queue name field is blank so that you can supply the name for the new queue. The description is that of the queue upon which you are basing this new definition. Over type this field with your own description for the new queue.

The values in the other fields are those of the queue upon which you are basing this new queue, except the disposition. You can over type these fields as you require. For example, type Y in the **Put enabled** field (if it is not already Y) if suitably authorized applications can put messages on this queue.

You get field help by moving the cursor into a field and pressing function key F1. Field help provides information about the values that can be used for each attribute.

When you have completed the first panel, press function key F8 to display the second panel.

#### Hints:

1. Do not press Enter at this stage, otherwise the queue will be created before you have a chance to complete the remaining fields. (If you do press Enter prematurely, do not worry; you can always alter your definition later on.)
2. Do not press function keys F3 or F12, or the data you typed will be lost.

Press function key F8 repeatedly to see and complete the remaining panels, including the trigger definition, event control, and backout reporting panels.

#### When your local queue definition is complete

When your definition is complete, press Enter to send the information to the queue manager for processing. The queue manager creates the queue according to the definition you have supplied. If you do not want the queue to be created, press function key F3 to exit and cancel the definition.

### Defining other types of objects

To define other types of object, use an existing definition as the base for your new definition as explained in [Defining a local queue](#).

Use the **Define like** action either on the initial panel or against an object entry in a list displayed as a result of options selected on the initial panel.

For example, starting from the initial panel, complete these fields:

<b>Action</b>	2 (Define like)
<b>Object type</b>	QALIAS, NAMELIST, PROCESS, CHANNEL, and other resource objects.
<b>Name</b>	Leave blank or enter the name of an existing object of the same type.

Press Enter to display the corresponding DEFINE panels. Complete the fields as required and then press Enter again to send the information to the queue manager.

Like defining a local queue, defining another type of object generally requires several panels to be completed. Defining a namelist requires some additional work, as described in [“Working with namelists”](#) on page 498.

### Working with object definitions

When an object has been defined, you can specify an action in the **Action** field, to alter, display, or manage it.

In each case, you can either:

- Select the object you want to work with from a list displayed as a result of options selected on the initial panel. For example, having entered 1 in the **Action** field to display objects, Queue in the **Object type** field, and \* in the **Name** field, you are presented with a list of all queues defined in the system. You can then select from this list the queue with which you need to work.

- Start from the initial panel, where you specify the object you are working with by completing the **Object type** and **Name** fields.

#### **Altering an object definition**

To alter an object definition, specify action 3 and press Enter to see the ALTER panels. These panels are very similar to the DEFINE panels. You can alter the values you want. When your changes are complete, press Enter to send the information to the queue manager.

#### **Displaying an object definition**

If you want to see the details of an object without being able to change them, specify action 1 and press Enter to see the DISPLAY panels. Again, these panels are similar to the DEFINE panels except that you cannot change any of the fields. Change the object name to display details of another object.

#### **Deleting an object**

To delete an object, specify action 4 (Manage) and the **Delete** action is one of the actions presented on the resulting menu. Select the **Delete** action.

You are asked to confirm your request. If you press function key F3 or F12, the request is canceled. If you press Enter, the request is confirmed and passed to the queue manager. The object you specified is then deleted.

**Note:** You cannot delete most types of channel object unless the channel initiator is started.

### **Working with namelists**

When working with namelists, proceed as you would for other objects.

For the actions DEFINE LIKE or ALTER, press function key F11 to add names to the list or to change the names in the list. This involves working with the ISPF editor and all the normal ISPF edit commands are available. Enter each name in the namelist on a separate line.

When you use the ISPF editor in this way, the function key settings are the normal ISPF settings, and **not** those used by the other operations and control panels.

If you need to specify lowercase names in the list, specify CAPS(OFF) on the editor panel command line. When you do this, all the namelists that you edit in the future are in lowercase until you specify CAPS(ON).

When you have finished editing the namelist, press function key F3 to end the ISPF edit session. Then press Enter to send the changes to the queue manager.

**Attention:** If you do not press Enter at this stage but press function key F3 instead, you lose any updates that you have typed in.

## **Implementing the system using multiple cluster transmission queues**

It makes no difference if the channel is used in a single cluster, or an overlapping cluster. When the channel is selected and started, the channel selects the transmission queue depending on the definitions.

### **Procedure**

- If you are using the DEFCLXQ option, see [“Using the automatic definition of queues and switching” on page 498](#).
- If you are using a staged approach, see [“Changing your cluster-sender channels using a phased approach” on page 499](#).

## **Using the automatic definition of queues and switching**

Use this option if you are planning on using the DEFCLXQ option. There will be a queue created for every channel, and every new channel.

## Procedure

1. Review the definition of the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE and change the attributes if required.

This queue is defined in member SCSQPROC(csq4insx).

2. Create the SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE model queue.
3. Apply security policies for this model queue, and the SYSTEM.CLUSTER.TRANSMIT.\*\* queues.

For z/OS the channel initiator started task user ID needs:

- Control access to CLASS(MQADMIN) for

```
ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channelname
```

- Update access to CLASS(MQQUEUE) for

```
ssid.SYSTEM.CLUSTER.TRANSMIT.channelname
```

## Changing your cluster-sender channels using a phased approach

Use this option if you are planning on using a staged approach. This process allows you to move to the new cluster-sender channels at various times to suit the needs of your enterprise.

### Before you begin

- Identify your business applications, and which channels are used.
- For the queues you use, display the clusters they are in.
- Display the channels to show the connection names, the names of the remote queue managers, and which clusters the channel supports.

### About this task

- Create a transmission queue. On z/OS you might want to consider which page set you use for the queue.
- Set up security policy for the queue.
- Change any queue monitoring to include this queue name.
- Decide which channels are to use this transmission queue. The channels should have a similar name, so generic characters ' \* ' in the CLCHNAME identify the channel.
- When you are ready to use the new function, alter the transmission queue to specify the name of the channels to use this transmission queue. For example CLUSTER1.TOPARIS, or CLUSTER1.\* or \*.TOPARIS
- Start the channels

## Procedure

1. Use the DIS CLUSQMGR(yyyy) XMITQ command to display the cluster sender channels defined in the cluster, where yyyy is the name of the remote queue manager.
2. Set up the security profile for the transmission queue and give the queue access to the channel initiator.
3. Define the transmission queue to be used, and specify USAGE(XMITQ) INDXTYPE( CORRELID ) SHARE and CLCHNAME(value)

The channel initiator started task user ID needs the following access:

```
alter class(MQADMIN) ssid.CONTEXT.SYSTEM.CLUSTER.TRANSMIT.channel  
update class(MQQUEUE) ssid.SYSTEM.CLUSTER.TRANSMIT.channel
```

and the user ID using the SWITCH command needs the following access:

```
alter cl(MQADMIN) ssid.QUEUE.queueName
```

#### 4. Stop and restart the channels.

The channel change occurs when the channel starts using an MQSC command, or you use CSQUTIL. You can identify which channels need to be restarted using the SWITCH CHANNEL (\*) STATUS of CSQUTIL

If you have problems when the channel is started, stop the channel, resolve the problems, and restart the channel.

Note that you can change the CLCHNAME attribute as often as you need to.

The value of CLCHNAME used is the one when the channel is started, so you can change the CLCHNAME definition while the channel continues to use the definitions from the time that it started. The channel uses the new definition when it is restarted.

## Undoing a change to a transmission queue on z/OS

You need to have a process to backout a change if the results are not as you expect.

### What can go wrong?

If the new transmission queue is not what you expect:

1. Check the CLCHNAME is as you expect
2. Review the job log to check if the switch process has finished. If not, wait and check the new transmission queue of the channel later.

If you are using multiple cluster transmission queues, it is important that you design the transmission queues definitions explicitly and avoid complicated overlapping configuration. In this way, you can make sure that if there are problems, you can go back to the original queues and configuration.

If you encounter problems during the move to using a different transmission queue, you must resolve any problems before you can proceed with the change.

An existing change request must complete before a new change request can be made. For example, you:

1. Define a new transmission queue with a maximum depth of one and there are 10 messages waiting to be sent.
2. Change the transmission queue to specify the channel name in the CLCHNAME parameter.
3. Stop and restart the channel. The attempt to move the messages fails and reports the problems.
4. Change the CLCHNAME parameter on the transmission queue to be blank.
5. Stop and restart the channel. The channel continues to try and complete the original request, so the channel continues to use the new transmission queue.
6. Need to resolve the problems and restart the channel so the moving of messages completes successfully.

Next time the channel is restarted it picks up any changes, so if you had set CLCHNAME to blanks, the channel will not use the specified transmission queue.

In this example, changing the CLCHNAME on the transmission queue to blanks does not necessarily mean that the channel uses the SYSTEM.CLUSTER.TRANSMIT queue, as there might be other transmission queues whose CLCHNAME parameter match the channel name. For example, a generic name, or the queue manager attribute DEFCLXQ might be set to channel, so the channel uses a dynamic queue instead of the SYSTEM.CLUSTER.TRANSMIT queue.

## Writing programs to administer IBM MQ for z/OS

You can write your own application programs to administer a queue manager. Use this topic to understand the requirements for writing your own administration programs.

### Start of General-use programming interface information

This set of topics contains hints and guidance to enable you to issue IBM MQ commands from an IBM MQ application program.

**Note:** In this topic, the MQI calls are described using C-language notation. For typical invocations of the calls in the COBOL, PL/I, and assembler languages, see [Function calls](#).

### Understanding how it all works

In outline, the procedure for issuing commands from an application program is as follows:

1. Build an IBM MQ command into a type of IBM MQ message called a *request message*. The command can be in MQSC or PCF format.
2. Send (use MQPUT ) this message to a special queue called the system-command input queue. The IBM MQ command processor runs the command.
3. Retrieve (use MQGET ) the results of the command as *reply messages* on the reply-to queue. These messages contain the user messages that you need to determine whether your command was successful and, if it was, what the results were.

Then it is up to your application program to process the results.

This set of topics contains:

## Preparing queues for administration programs

Administration programs require a number of predefined queues for system command input and receiving responses.

This section applies to commands in the MQSC format. For the equivalent in PCF, see [“Utilisation des formats de commande programmables IBM MQ”](#) on page 28.

Before you can issue any MQPUT or MQGET calls, you must first define, and then open, the queues you are going to use.

### Defining the system-command input queue

The system-command input queue is a local queue called SYSTEM.COMMAND.INPUT. The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a default definition for the system-command input queue. For compatibility with IBM MQ on other platforms, an alias of this queue, called SYSTEM.ADMIN.COMMAND.QUEUE is also supplied. See [Sample definitions supplied with IBM MQ](#) for more information.

### Defining a reply-to queue

You must define a reply-to queue to receive reply messages from the IBM MQ command processor. It can be any queue with attributes that allow reply messages to be put on it. However, for normal operation, specify these attributes:

- USAGE(NORMAL)
- NOTRIGGER (unless your application uses triggering)

Avoid using persistent messages for commands, but if you choose to do so, the reply-to queue must not be a temporary dynamic queue.

The supplied CSQINP2 initialization data set, thlqual.SCSQPROC(CSQ4INSG), contains a definition for a model queue called SYSTEM.COMMAND.REPLY.MODEL. You can use this model to create a dynamic reply-to queue.

**Note:** Replies generated by the command processor can be up to 15 000 bytes in length.

If you use a permanent dynamic queue as a reply-to queue, your application should allow time for all PUT and GET operations to complete before attempting to delete the queue, otherwise MQRC2055 (MQRC\_Q\_NOT\_EMPTY) can be returned. If this occurs, try the queue deletion again after a few seconds.

## Opening the system-command input queue

Before you can open the system-command input queue, your application program must be connected to your queue manager. Use the MQI call MQCONN or MQCONNX to do this.

Then use the MQI call MQOPEN to open the system-command input queue. To use this call:

1. Set the **Options** parameter to MQOO\_OUTPUT
2. Set the MQOD object descriptor fields as follows:

**ObjectType**

MQOT\_Q (the object is a queue)

**ObjectName**

SYSTEM.COMMAND.INPUT

**ObjectQMgrName**

If you want to send your request messages to your local queue manager, leave this field blank. This means that your commands are processed locally.

If you want your IBM MQ commands to be processed on a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

## Opening a reply-to queue

To retrieve the replies from an IBM MQ command, you must open a reply-to queue. One way of doing this is to specify the model queue, SYSTEM.COMMAND.REPLY.MODEL in an MQOPEN call, to create a permanent dynamic queue as the reply-to queue. To use this call:

1. Set the **Options** parameter to MQOO\_INPUT\_SHARED
2. Set the MQOD object descriptor fields as follows:

**ObjectType**

MQOT\_Q (the object is a queue)

**ObjectName**

The name of the reply-to queue. If the queue name you specify is the name of a model queue object, the queue manager creates a dynamic queue.

**ObjectQMgrName**

To receive replies on your local queue manager, leave this field blank.

**DynamicQName**

Specify the name of the dynamic queue to be created.

## Using the command server

The command server is an IBM MQ component that works with the command processor component. You can send formatted messages to the command server which interprets the messages, runs the administration requests, and sends responses back to your administration application.

The command server reads request messages from the system-command input queue, verifies them, and passes the valid ones as commands to the command processor. The command processor processes the commands and puts any replies as reply messages on to the reply-to queue that you specify. The first reply message contains the user message CSQN205I. See [“Interpreting the reply messages from the command server”](#) on page 506 for more information. The command server also processes channel initiator and queue sharing group commands, wherever they are issued from.

### Identifying the queue manager that processes your commands

The queue manager that processes the commands you issue from an administration program is the queue manager that owns the system-command input queue that the message is put onto.

### Starting the command server

Normally, the command server is started automatically when the queue manager is started. It becomes available as soon as the message CSQ9022I 'START QMGR' NORMAL COMPLETION is returned from the START QMGR command. The command server is stopped when all the connected tasks have been disconnected during the system termination phase.

You can control the command server yourself using the START CMDSERV and STOP CMDSERV commands. To prevent the command server starting automatically when IBM MQ is restarted, you can add a STOP CMDSERV command to your CSQINP1 or CSQINP2 initialization data sets. However, this is not recommended as it prevents any channel initiator or queue sharing group commands being processed.

The STOP CMDSERV command stops the command server as soon as it has finished processing the current message, or immediately if no messages are being processed.

If the command server has been stopped by a STOP CMDSERV command in the program, no other commands from the program can be processed. To restart the command server, you must issue a START CMDSERV command from the z/OS console.

If you stop and restart the command server while the queue manager is running, all the messages that are on the system-command input queue when the command server stops are processed when the command server is restarted. However, if you stop and restart the queue manager after the command server is stopped, only the persistent messages on the system-command input queue are processed when the command server is restarted. All nonpersistent messages on the system-command input queue are lost.

### Sending commands to the command server

For each command, you build a message containing the command, then put it onto the system-command input queue.

### Building a message that includes IBM MQ commands

You can incorporate IBM MQ commands in an application program by building request messages that include the required commands. For each such command you:

1. Create a buffer containing a character string representing the command.
2. Issue an MQPUT call specifying the buffer name in the **buffer** parameter of the call.

The simplest way to do this in C is to define a buffer using 'char'. For example:

```
char message_buffer[ ] = "ALTER QLOCAL(SALES) PUT(ENABLED)";
```

When you build a command, use a null-terminated character string. Do not specify a command prefix string (CPF) at the start of a command defined in this way. This means that you do not have to alter your command scripts if you want to run them on another queue manager. However, you must take into account that a CPF is included in any response messages that are put onto the reply-to queue.

The command server folds all lowercase characters to uppercase unless they are inside quotation marks.

Commands can be any length up to a maximum 32 762 characters.

### Putting messages on the system-command input queue

Use the MQPUT call to put request messages containing commands on the system-command input queue. In this call you specify the name of the reply-to queue that you have already opened.

To use the MQPUT call:

1. Set these MQPUT parameters:

**Hconn**

The connection handle returned by the MQCONN or MQCONNX call.

**Hobj**

The object handle returned by the MQOPEN call for the system-command input queue.

**BufferLength**

The length of the formatted command.

**Buffer**

The name of the buffer containing the command.

2. Set these MQMD fields:

**MsgType**

MQMT\_REQUEST

**Format**

MQFMT\_STRING or MQFMT\_NONE

If you are not using the same code page as the queue manager, set *CodedCharSetId* as appropriate and set MQFMT\_STRING, so that the command server can convert the message. Do not set MQFMT\_ADMIN, as that causes your command to be interpreted as PCF.

**ReplyToQ**

Name of your reply-to queue.

**ReplyToQMgr**

If you want replies sent to your local queue manager, leave this field blank. If you want your IBM MQ commands to be sent to a remote queue manager, put its name here. You must also have the correct queues and links set up, as described in [Distributed queuing and clusters](#).

3. Set any other MQMD fields, as required. You should normally use nonpersistent messages for commands.

4. Set any *PutMsgOpts* options, as required.

If you specify MQPMO\_SYNCPOINT (the default), you must follow the MQPUT call with a syncpoint call.

### Using MQPUT1 and the system-command input queue

If you want to put just one message on the system-command input queue, you can use the **MQPUT1** call. This call combines the functions of an **MQOPEN**, followed by an **MQPUT** of one message, followed by an **MQCLOSE**, all in one call. If you use this call, modify the parameters accordingly. See [Putting one message on a queue using the MQPUT1 call](#) for details.

## Retrieving replies to your commands

The command server sends a response to a reply queue for each request message it receives. Any administration application must receive, and handle the reply messages.

When the command processor processes your commands, any reply messages are put onto the reply-to queue specified in the MQPUT call. The command server sends the reply messages with the same persistence as the command message it received.



## Waiting for a reply

Use the MQGET call to retrieve a reply from your request message. One request message can produce several reply messages. For details, see [“Interpreting the reply messages from the command server” on page 506](#).

You can specify a time interval that an MQGET call waits for a reply message to be generated. If you do not get a reply, use the checklist beginning in topic [“If you do not receive a reply” on page 507](#).

To use the MQGET call:

1. Set these parameters:

**Hconn**

The connection handle returned by the MQCONN or MQCONNX call.

**Hobj**

The object handle returned by the MQOPEN call for the reply-to queue.

**Buffer**

The name of the area to receive the reply.

**BufferLength**

The length of the buffer to receive the reply. This must be a minimum of 80 bytes.

2. To ensure that you only get the responses from the command that you issued, you must specify the appropriate *MsgId* and *CorrelId* fields. These depend on the report options, MQMD\_REPORT, you specified in the MQPUT call:

**MQRO\_NONE**

Binary zero, '00...00' (24 nulls).

**MQRO\_NEW\_MSG\_ID**

Binary zero, '00...00' (24 nulls).

This is the default if none of these options has been specified.

**MQRO\_PASS\_MSG\_ID**

The *MsgId* from the MQPUT.

**MQRO\_NONE**

The *MsgId* from the MQPUT call.

**MQRO\_COPY\_MSG\_ID\_TO\_CORREL\_ID**

The *MsgId* from the MQPUT call.

This is the default if none of these options has been specified.

**MQRO\_PASS\_CORREL\_ID**

The *CorrelId* from the MQPUT call.

For more details on report options, see [Report options and message flags](#).

3. Set the following *GetMsgOpts* fields:

**Options**

MQGMO\_WAIT

If you are not using the same code page as the queue manager, set MQGMO\_CONVERT, and set *CodedCharSetId* as appropriate in the MQMD.

**WaitInterval**

For replies from the local queue manager, try 5 seconds. Coded in milliseconds, this becomes 5 000. For replies from a remote queue manager, and channel control and status commands, try 30 seconds. Coded in milliseconds, this becomes 30 000.

## Discarded messages

If the command server finds that a request message is not valid, it discards this message and writes the message CSQN205I to the named reply-to queue. If there is no reply-to queue, the CSQN205I message is put onto the dead-letter queue. The return code in this message shows why the original request message was not valid:

- 00D5020F** It is not of type MQMT\_REQUEST.
- 00D50210** It has zero length.
- 00D50212** It is longer than 32 762 bytes.
- 00D50211** It contains all blanks.
- 00D5483E** It needed converting, but *Format* was not MQFMT\_STRING.
- Other** See [Command server codes](#)

### The command server reply message descriptor

For any reply message, the following MQMD message descriptor fields are set:

<i>MsgType</i>	MQMT_REPLY
<i>Feedback</i>	MQFB_NONE
<i>Encoding</i>	MQENC_NATIVE
<i>Priority</i>	As for the MQMD in the message you issued.
<i>Persistence</i>	As for the MQMD in the message you issued.
<i>CorrelId</i>	Depends on the MQPUT report options.
<i>ReplyToQ</i>	None.

The command server sets the *Options* field of the MQPMO structure to MQPMO\_NO\_SYNCPOINT. This means that you can retrieve the replies as they are created, rather than as a group at the next syncpoint.

## Interpreting the reply messages from the command server

Each request message correctly processed by IBM MQ produces at least two reply messages. Each reply message contains a single IBM MQ user message.

The length of a reply depends on the command that was issued. The longest reply you can get is from a **DISPLAY NAMELIST** command, and that can be up to 15 000 bytes in length.

The first user message, CSQN205I, always contains:

- A count of the replies (in decimal), which you can use as a counter in a loop to get the rest of the replies. The count includes this first message.
- The return code from the command preprocessor.
- A reason code, which is the reason code from the command processor.

This message does not contain a CPF.

For example:

```
CSQN205I    COUNT=      4, RETURN=0000000C, REASON=00000008
```

The COUNT field is 8 bytes long and is right-justified. It always starts at position 18, that is, immediately after COUNT=. The RETURN field is 8 bytes long in character hexadecimal and is immediately after RETURN= at position 35. The REASON field is 8 bytes long in character hexadecimal and is immediately after REASON= at position 52.

If the RETURN= value is 00000000 and the REASON= value is 00000004, the set of reply messages is incomplete. After retrieving the replies indicated by the CSQN205I message, issue a further MQGET call to wait for a further set of replies. The first message in the next set of replies is again CSQN205I, indicating how many replies there are, and whether there are still more to come.

See the [Messages, codes achèvement et codes anomalie IBM MQ for z/OS](#) documentation for more details about the individual messages.

If you are using a non-English language feature, the text and layout of the replies are different from those shown here. However, the size and position of the count and return codes in message CSQN205I are the same.

## **If you do not receive a reply**

There are a series of steps you can take if you do not receive a response to request to the command server.

If you do not receive a reply to your request message, work through this checklist:

- Is the command server running?
- Is the *WaitInterval* long enough?
- Are the system-command input and reply-to queues correctly defined?
- Were the MQOPEN calls to these queues successful?
- Are both the system-command input and reply-to queues enabled for MQPUT and MQGET calls?
- Have you considered increasing the MAXDEPTH and MAXMSGL attributes of your queues?
- Are you are using the *CorrelId* and *MsgId* fields correctly?
- Is the queue manager still running?
- Was the command built correctly?
- Are all your remote links defined and operating correctly?
- Were the MQPUT calls correctly defined?
- Has the reply-to queue been defined as a temporary dynamic queue instead of a permanent dynamic queue? (If the request message is persistent, you must use a permanent dynamic queue for the reply.)

When the command server generates replies but cannot write them to the reply-to queue that you specify, it writes them to the dead-letter queue.

## **Passing commands using MGCRE**

With appropriate authorization, an application program can make requests to multiple queue managers using a z/OS service routine.

If you have the correct authorization, you can pass IBM MQ commands from your program to multiple queue managers by the MGCRE (SVC 34) z/OS service. See the [z/OS MVS Programming: Authorized Assembler Services Guide](#) for more information.

The value of the CPF identifies the particular queue manager to which the command is directed. For information about CPFs, see [User IDs for command security and command resource security](#) and [“Issuing queue manager commands on z/OS” on page 470](#).

If you use MGCRE, you can use a Command and Response Token (CART) to get the direct responses to the command.

## **Examples of commands and their replies**

Use this topic as a series of examples of commands to the command server and the responses from the command server.

Here are some examples of commands that could be built into IBM MQ messages, and the user messages that are the replies. Unless otherwise stated, each line of the reply is a separate message.

- [Messages from a DEFINE command](#)
- [Messages from a DELETE command](#)
- [Messages from DISPLAY commands](#)
- [Messages from commands with CMDSCOPE](#)
- [Messages from commands that generate commands with CMDSCOPE](#)

### Messages from a DEFINE command

The following command:

```
DEFINE QLOCAL(Q1)
```

produces these messages:

```
CSQN205I    COUNT=    2, RETURN=00000000, REASON=00000000  
CSQ9022I +CSQ1 CSQMMSGP ' DEFINE QLOCAL' NORMAL COMPLETION
```

These reply messages are produced on normal completion.

### Messages from a DELETE command

The following command:

```
DELETE QLOCAL(Q2)
```

produces these messages:

```
CSQN205I    COUNT=    4, RETURN=00000000, REASON=00000000  
CSQM125I +CSQ1 CSQMUQLC QLOCAL (Q2) QSGDISP(QMGR) WAS NOT FOUND  
CSQM090E +CSQ1 CSQMUQLC FAILURE REASON CODE X'00D44002'  
CSQ9023E +CSQ1 CSQMUQLC ' DELETE QLOCAL' ABNORMAL COMPLETION
```

These messages indicate that a local queue called Q2 does not exist.

### Messages from DISPLAY commands

The following examples show the replies from some DISPLAY commands.

#### Finding out the name of the dead-letter queue

If you want to find out the name of the dead-letter queue for a queue manager, issue this command from an application program:

```
DISPLAY QMGR DEADQ
```

The following three user messages are returned, from which you can extract the required name:

```

CSQN205I   COUNT=    3, RETURN=00000000, REASON=00000000
CSQM409I +CSQ1 QMNAME(CSQ1) DEADQ(SYSTEM.DEAD.QUEUE      )
CSQ9022I +CSQ1 CSQMDRTS ' DISPLAY QMGR' NORMAL COMPLETION

```

## Messages from the DISPLAY QUEUE command

The following examples show how the results from a command depend on the attributes specified in that command.

### Example 1

You define a local queue using the command:

```

DEFINE QLOCAL(Q1) DESCR('A sample queue') GET(ENABLED) SHARE

```

If you issue the following command from an application program:

```

DISPLAY QUEUE(Q1) SHARE GET DESCR

```

these three user messages are returned:

```

CSQN205I   COUNT=    3, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(Q1                                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
DESCR(A sample queue
) SHARE GET(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION

```

**Note:** The second message, CSQM401I, is shown here occupying four lines.

### Example 2

Two queues have names beginning with the letter A:

- A1 is a local queue with its PUT attribute set to DISABLED.
- A2 is a remote queue with its PUT attribute set to ENABLED.

If you issue the following command from an application program:

```

DISPLAY QUEUE(A*) PUT

```

these four user messages are returned:

```

CSQN205I   COUNT=    4, RETURN=00000000, REASON=00000000
CSQM401I +CSQ1 QUEUE(A1                                ) TYPE(
QLOCAL ) QSGDISP(QMGR  )
PUT(DISABLED )
CSQM406I +CSQ1 QUEUE(A2                                ) TYPE(
QREMOTE ) PUT(ENABLED )
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY QUEUE' NORMAL COMPLETION

```

**Note:** The second and third messages, CSQM401I and CSQM406I, are shown here occupying three and two lines.

### Messages from the DISPLAY NAMELIST command

You define a namelist using the command:

```
DEFINE NAMELIST(N1) NAMES(Q1,SAMPLE_QUEUE)
```

If you issue the following command from an application program:

```
DISPLAY NAMELIST(N1) NAMES NAMCOUNT
```

the following three user messages are returned:

```
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000000
CSQM407I +CSQ1 NAMELIST(N1
GDISP(QMGR ) NAMCOUNT(    2) NAMES(Q1
,SAMPLE_QUEUE
CSQ9022I +CSQ1 CSQMMSG ' DISPLAY NAMELIST' NORMAL COMPLETION
```

**Note:** The second message, CSQM407I, is shown here occupying three lines.

### Messages from commands with CMDSCOPE

The following examples show the replies from commands that have been entered with the CMDSCOPE attribute.

#### Messages from the ALTER PROCESS command

The following command:

```
ALT PRO(V4) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'ALT PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I  COUNT=    5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ26
CSQM125I !MQ26 CSQMMSGP PROCESS(V4) QSGDISP(QMGR) WAS NOT FOUND
CSQM090E !MQ26 CSQMMSGP FAILURE REASON CODE X'00D44002'
CSQ9023E !MQ26 CSQMMSGP ' ALT PRO' ABNORMAL COMPLETION
CSQN205I  COUNT=    3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'ALT PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' ALT PRO' NORMAL COMPLETION
CSQN205I  COUNT=    2, RETURN=00000000, REASON=00000008
CSQN123E !MQ25 'ALT PRO' command for CMDSCOPE(*) abnormal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). The command was successful on MQ25 but the process definition did not exist on MQ26, so the command failed on that queue manager.

#### Messages from the DISPLAY PROCESS command

The following command:

```
DIS PRO(V*) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS PRO' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 5, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ26
CSQM408I !MQ26 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ26 PROCESS(V3) QSGDISP(QMGR)
CSQ9022I !MQ26 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 7, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS PRO' command responses from MQ25
CSQM408I !MQ25 PROCESS(V2) QSGDISP(COPY)
CSQM408I !MQ25 PROCESS(V2) QSGDISP(GROUP)
CSQM408I !MQ25 PROCESS(V3) QSGDISP(QMGR)
CSQM408I !MQ25 PROCESS(V4) QSGDISP(QMGR)
CSQ9022I !MQ25 CSQMDRTS 'DIS PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS PRO' command for CMDSCOPE(*) normal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about all the processes on each queue manager with names starting with the letter V.

### Messages from the DISPLAY CHSTATUS command

The following command:

```
DIS CHS(VT) CMDSCOPE(*)
```

produces the following messages:

```
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'DIS CHS' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ25
CSQM422I !MQ25 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ25 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 4, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DIS CHS' command responses from MQ26
CSQM422I !MQ26 CHSTATUS(VT) CHLDISP(PRIVATE) CONNAME( ) CURRENT STATUS(STOPPED)
CSQ9022I !MQ26 CSQXDRTS 'DIS CHS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DIS CHS' command for CMDSCOPE(*) normal completion
```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Information is displayed about channel status on each queue manager.

### Messages from the STOP CHANNEL command

The following command:

```
STOP CHL(VT) CMDSCOPE(*)
```

produces these messages:

```

CSQN205I COUNT= 2, RETURN=00000000, REASON=00000004
CSQN137I !MQ25 'STOP CHL' command accepted for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQM134I !MQ25 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
SQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQM134I !MQ26 CSQMTCHL STOP CHL(VT) COMMAND ACCEPTED
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ26
CSQ9022I !MQ26 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'STOP CHL' command responses from MQ25
CSQ9022I !MQ25 CSQXCRPS ' STOP CHL' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'STOP CHL' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25 and sent to two queue managers (MQ25 and MQ26). Channel VT was stopped on each queue manager.

### Messages from commands that generate commands with CMDSCOPE

The following command:

```
DEF PRO(V2) QSGDISP(GROUP)
```

produces these messages:

```

CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQM122I !MQ25 CSQMMSGP ' DEF PRO' COMPLETED FOR QSGDISP(GROUP)
CSQN138I !MQ25 'DEFINE PRO' command generated for CMDSCOPE(*), sent to 2
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ25
CSQ9022I !MQ25 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 3, RETURN=00000000, REASON=00000004
CSQN121I !MQ25 'DEFINE PRO' command responses from MQ26
CSQ9022I !MQ26 CSQMMSGP ' DEFINE PROCESS' NORMAL COMPLETION
CSQN205I COUNT= 2, RETURN=00000000, REASON=00000000
CSQN122I !MQ25 'DEFINE PRO' command for CMDSCOPE(*) normal completion

```

These messages tell you that the command was entered on queue manager MQ25. When the object was created on the shared repository, another command was generated and sent to all the active queue managers in the queue sharing group (MQ25 and MQ26).

**z/OS**

## Managing IBM MQ resources on z/OS

Use the links in this topic to find out how to manage the resources used by IBM MQ for z/OS, for example, managing log files, data sets, page sets, buffer pools, and coupling facility structures.

Use the following links for details of the different administrative tasks you might have to complete while using IBM MQ for z/OS:

- [“Managing the logs” on page 513](#)
- [“Managing the bootstrap data set \(BSDS\)” on page 522](#)
- [“Managing page sets” on page 529](#)
- [“How to back up and recover page sets” on page 536](#)
- [“How to back up and restore queues using CSQUTIL” on page 539](#)
- [“Managing buffer pools” on page 539](#)



- [“Managing queue sharing groups and shared queues on z/OS” on page 541](#)

### Related concepts

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

[“Administering IBM MQ for z/OS” on page 469](#)

IBM MQ for z/OS can be controlled and managed by MQSC and PCF commands, by a set of utilities and programs provided with the product, and by authorized applications.

[“Sources from which you can issue MQSC and PCF commands on IBM MQ for z/OS” on page 470](#)

You can issue MQSC and PCF commands from the IBM MQ for z/OS console, the initialization input data sets, the batch utility CSQUTIL, or from authorized applications. Not all commands can be issued from all these sources.

[“Recovery and restart on z/OS” on page 550](#)

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

### Related tasks

[Planning your IBM MQ environment on z/OS](#)

[Configuring queue managers on z/OS](#)

[IBM MQ utilities on z/OS reference](#)

### Related reference

[“Using the IBM MQ for z/OS utilities” on page 492](#)

IBM MQ for z/OS provides a set of utility programs that you can use to help with system administration.

[Programmable command formats reference](#)

## Managing the logs

Use this topic to understand how to manage your IBM MQ log files, including the log archiving process, using log record compression, log record recovery, and printing log records.

This topic describes the tasks involved in managing the IBM MQ logs. It contains these sections:

### Archiving logs with the ARCHIVE LOG command

An authorized operator can archive the current IBM MQ active log data sets whenever required using the **ARCHIVE LOG** command.

When you issue the ARCHIVE LOG command, IBM MQ truncates the current active log data sets, then runs an asynchronous offload process, and updates the BSDS with a record of the offload process.

The **ARCHIVE LOG** command has a **MODE(QUIESCE)** option. With this option, IBM MQ jobs and users are quiesced after a commit point, and the resulting point of consistency is captured in the current active log before it is offloaded.

Consider using the **MODE(QUIESCE)** option when planning a backup strategy for off site recovery. It creates a system-wide point of consistency, which minimizes the number of data inconsistencies when the archive log is used with the most current backup page set copy during recovery. For example:

```
ARCHIVE LOG MODE(QUIESCE)
```

If you issue the **ARCHIVE LOG** command without specifying a **TIME** parameter, the quiesce time period defaults to the value of the **QUIESCE** parameter of the CSQ6ARVP macro. If the time required for the ARCHIVE LOG MODE(QUIESCE) to complete is less than the time specified, the command completes successfully; otherwise, the command fails when the time period expires. You can specify the time period explicitly by using the **TIME** option, for example:

```
ARCHIVE LOG MODE(QUIESCE) TIME(60)
```

This command specifies a quiesce period of up to 60 seconds before **ARCHIVE LOG** processing occurs.

**Attention:** Using the **TIME** option when time is critical can significantly disrupt IBM MQ availability for all jobs and users that use IBM MQ resources.

By default, the command is processed asynchronously from the time you submit the command. (To process the command synchronously with other IBM MQ commands use the **WAIT (YES)** option with **QUIESCE**, but be aware that the z/OS console is locked from IBM MQ command input for the entire **QUIESCE** period.)

During the quiesce period:

- Jobs and users on the queue manager are allowed to go through commit processing, but are suspended if they try to update any IBM MQ resource after the commit.
- Jobs and users that only read data can be affected, since they might be waiting for locks held by jobs or users that were suspended.
- New tasks can start, but they cannot update data.

The output from the **DISPLAY LOG** command uses the message CSQV400I to indicate that a quiesce is in effect.

For example:

```
CSQJ322I +CSQ1 DISPLAY LOG report ...
Parameter  Initial value      SET value
-----
INBUFF     60
OUTBUFF    400
MAXRTU     2
MAXARCH    2
TWOACTV    YES
TWOARCH    YES
TWOBSDS    YES
OFFLOAD    YES
MAXCNOFF   0
WRTHRS    20
DEALLCT    0
COMPLG     NONE
ZHYWRITE   NO
End of LOG report
CSQJ370I +CSQ1 LOG status report ...
Copy %Full zHyperWrite Encrypted DSName
  1   68 NO          NO          VICY.CSQ1.LOGCOPY1.DS01
  2   68 NO          NO          VICY.CSQ1.LOGCOPY2.DS01
Restarted at 2019-08-15 09:49:30 using RBA=000000000891B000
Latest RBA=000000000891CCF8
Offload task is AVAILABLE
Full logs to offload - 0 of 4
CSQV400I +CSQ1 ARCHIVE LOG QUIESCE CURRENTLY ACTIVE
CSQ9022I +CSQ1 CSQJC001 ' DISPLAY LOG' NORMAL COMPLETION
```

When all updates are quiesced, the quiesce history record in the BSDS is updated with the date and time that the active log data sets were truncated, and with the last-written RBA in the current active log data sets. IBM MQ truncates the current active log data sets, switches to the next available active log data sets, and issues message CSQJ311I stating that the offload process started.

If updates cannot be quiesced before the quiesce period expires, IBM MQ issues message CSQJ317I, and **ARCHIVE LOG** processing terminates. The current active log data sets are not truncated, nor switched to the next available log data sets, and the offload process is not started.

Whether the quiesce was successful or not, all suspended users and jobs are then resumed, and IBM MQ issues message CSQJ312I, stating that the quiesce is ended and update activity is resumed.

If **ARCHIVE LOG** is issued when the current active log is the last available active log data set, the command is not processed, and IBM MQ issues the following message:

```
CSQJ319I - csect-name CURRENT ACTIVE LOG DATA SET IS THE LAST
AVAILABLE ACTIVE LOG DATA SET. ARCHIVE LOG PROCESSING
WILL BE TERMINATED
```

If **ARCHIVE LOG** is issued when another **ARCHIVE LOG** command is already in progress, the new command is not processed, and IBM MQ issues the following message:

CSQJ318I - ARCHIVE LOG COMMAND ALREADY IN PROGRESS

For information about the messages issued during archiving, see [Messages for IBM MQ for z/OS](#).

## Restarting the log archive process after a failure

If there is a problem during the log archive process (for example, a problem with allocation or tape mounts), the archiving of the active log might be suspended. You can cancel the archive process and restart it by using the following command:

```
ARCHIVE LOG CANCEL OFFLOAD
```

This command cancels any offload processing currently in progress, and restarts the archive process. It starts with the oldest log data set that has not been archived, and proceeds through all active log data sets that need offloading. Any log archive operations that have been suspended are restarted.

Use this command only if you are sure that the current log archive task is no longer functioning, or if you want to restart a previous attempt that failed. This is because the command might cause an abnormal termination of the offload task, which might result in a dump.

## Controlling archiving and logging

You can control compression, printing, archiving, recovery and logging with using the CSQ6LOGP, CSQ6ARVP, and CSQ6SYSP macros. Note, that changes to private objects only are logged in IBM MQlogs. Changes to GROUP objects (like shared inbound channels) are also logged, because the definitions are propagated around the group and held locally.

Many aspects of archiving and logging are controlled by parameters set using the CSQ6LOGP, CSQ6ARVP and CSQ6SYSP macros of the system parameter module when the queue manager is customized. See [Tailor your system parameter module](#) for details of these macros.

Some of these parameters can be changed while a queue manager is running using the IBM MQ MQSC SET LOG, SET SYSTEM and SET ARCHIVE commands. They are shown in [Table 28 on page 515](#):

SET command	Parameters
LOG	WRTHRSH, MAXARCH, DEALLCT, MAXRTU, COMPLOG
ARCHIVE	All
SYSTEM	LOGLOAD

You can display the settings of all the parameters using the MQSC [DISPLAY LOG](#), [DISPLAY ARCHIVE](#) and [DISPLAY SYSTEM](#) commands. These commands also show status information about archiving and logging.

## Controlling log compression

You can enable and disable the compression of log records using either

- The SET and DISPLAY LOG commands in MQSC; see [The MQSC commands](#)
- Invoking PCF interface. See [“Introduction à IBM MQ Programmable Command Formats” on page 27](#)
- Using the CSQ6LOGP macro in the system parameter module; see [Using CSQ6LOGP](#)

## Printing log records

You can extract and print log records using the CSQ1LOGP utility. For instructions, see [The log print utility](#).

## Recovering logs

Normally, you do not need to back up and restore the IBM MQ logs, especially if you are using dual logging. However, in rare circumstances, such as an I/O error on a log, you might need to recover the logs. Use Access Method Services to delete and redefine the data set, and then copy the corresponding dual log into it.

### **Discarding archive log data sets**

You can discard your archive log data sets and choose to discard the logs automatically or manually.

You must keep enough log data to be able to perform unit of work recovery, page set media recovery if a page set is lost, or CF structure media recovery if a CF structure is lost. Do not discard archive log data sets that might be required for recovery; if you discard these archive log data sets you might not be able to perform required recovery operations.

If you have confirmed that your archive log data sets can be discarded, you can do this in either of the following ways:

- [Automatic archive log data set deletion](#)
- [Manually deleting archive log data sets](#)

### **Automatic archive log data set deletion**

You can use a DASD or tape management system to delete archive log data sets automatically. The retention period for IBM MQ archive log data sets is specified by the retention period field ARCRETN in the CSQ6ARVP installation macro (see the [Using CSQ6ARVP](#) for more information).

The default for the retention period specifies that archive logs are to be kept for 9999 days (the maximum).

**Important:** You can change the retention period but you must ensure that you can accommodate the number of backup cycles that you have planned for.

.

IBM MQ uses the retention period value as the value for the JCL parameter RETPD when archive log data sets are created.

The retention period set by the MVS™/DFP storage management subsystem (SMS) can be overridden by this IBM MQ parameter. Typically, the retention period is set to the smaller value specified by either IBM MQ or SMS. The storage administrator and IBM MQ administrator must agree on a retention period value that is appropriate for IBM MQ.

**Note:** IBM MQ does not have an automated method to delete information about archive log data sets from the BSDS, because some tape management systems provide external manual overrides of retention periods. Therefore, information about an archive log data set can still be in the BSDS long after the data set retention period has expired and the data set has been scratched by the tape management system. Conversely, the maximum number of archive log data sets might have been exceeded and the data from the BSDS might have been dropped before the data set has reached its expiration date.

If archive log data sets are deleted automatically, remember that the operation does not update the list of archive logs in the BSDS. You can update the BSDS with the change log inventory utility, as described in “[Changing the BSDS](#)” on page 523. The update is not essential. Recording old archive logs wastes space in the BSDS, but does no other harm.

### **Manually deleting archive log data sets**

You must keep all the log records as far back as the lowest RBA identified in messages CSQI024I and CSQI025I. This RBA is obtained using the DISPLAY USAGE command that you issued when creating a point of recovery using [Method 1: Full backup](#).

### **Read [Creating a point of recovery for non-shared resources before discarding any logs](#).**

#### **Locate and discard archive log data sets**

Having established the minimum log RBA required for recovery, you can find archive log data sets that contain only earlier log records by performing the following procedure:

1. Use the print log map utility to print the contents of the BSDS. For an example of the output, see [The print log map utility](#).
2. Find the sections of the output titled ARCHIVE LOG COPY n DATA SETS. If you use dual logging, there are two sections. The columns labeled STARTRBA and ENDRBA show the range of RBAs contained in each volume. Find the volumes with ranges that include the minimum RBA you found with messages CSQI024I and CSQI025I. These are the earliest volumes you need to keep. If you are using dual-logging, there are two such volumes.

If no volumes have an appropriate range, one of the following cases applies:

- The minimum RBA has not yet been archived, and you can discard all archive log volumes.
- The list of archive log volumes in the BSDS wrapped around when the number of volumes exceeded the number allowed by the MAXARCH parameter of the CSQ6LOGP macro. If the BSDS does not register an archive log volume, that volume cannot be used for recovery. Therefore, consider adding information about existing volumes to the BSDS. For instructions, see [“Changes for archive logs” on page 525](#).

Also consider increasing the value of MAXARCH. For information, see the [Using CSQ6LOGP](#).

3. Delete any archive log data set or volume with an ENDRBA value that is less than the STARTRBA value of the earliest volume you want to keep. If you are using dual logging, delete both such copies.

Because BSDS entries wrap around, the first few entries in the BSDS archive log section might be more recent than the entries at the end. Look at the combination of date and time and compare their ages. Do not assume that you can discard all entries before the entry for the archive log containing the minimum LOGRBA.

Delete the data sets. If the archives are on tape, erase the tapes. If they are on DASD, run a z/OS utility to delete each data set. Then, if you want the BSDS to list only existing archive volumes, use the change log inventory utility (CSQJU003) to delete entries for the discarded volumes. See [“Changes for archive logs” on page 525](#) for an example.

#### **The effect of log shunting**

Long running transactions can cause unit of work log records which span log data sets. IBM MQ handles this scenario by using log shunting, a technique which moves the log records to optimize the quantity of log data retained, and queue manager restart time.

When a unit of work is considered to be long, a representation of each log record is written further down the log. This is known as *log shunting*. It is described more fully in [Log files](#).

The queue manager uses these shunted log records instead of the originals after a failure, to ensure unit of work integrity. There are two benefits to this:

- the quantity of log data which must be retained for unit of work coordination is reduced
- less log data must be traversed at queue manager restart time, so the queue manager is restarted more quickly

Shunted log records do not contain sufficient information for media recovery operations.

Data held in the log is used for two distinct purposes; media recovery and unit of work coordination. If a media failure occurs which affects either a CF structure or page set, the queue manager can recover the media to the point of failure by restoring a prior copy and updating this using data contained in the log.

Persistent activity performed in a unit of work is recorded on the log so that in the event of a failure, it can either be backed out or locks can be recovered on changed resources. The quantity of log data you need to retain to enable queue manager recovery is affected by these two elements.

For media recovery, you must retain sufficient log data to be able to perform media recovery from at least the most recent media copy and to be able to back out. (Your site may stipulate the ability to recover from older backups.) For unit of work integrity, you must retain the log data for your oldest in flight or indoubt units of work.

To assist you with managing the system, the queue manager detects old units of work at each log archive and reports them in messages CSQJ160 and CSQJ161. An internal task reads unit of work log information for these old units of work and rewrites it in a more succinct form to the current position in the log. Message CSQR026 indicates when this has happened. The MQSC command DISPLAY USAGE TYPE(DATASET) can also help you to manage the retention of log data. The command reports the following three pieces of recovery information:

1. How much of the log must be retained for unit of work recovery.
2. How much of the log must be retained for media recovery of page sets.
3. For a queue manager in a queue sharing group, how much of the log must be retained for media recovery of CF structures.

For each of these pieces of information, an attempt is made to map the oldest log data required into a data set. As new units of work start and stop, (1) would be expected to move to a more recent position in the log. If it is not moving, the long running UOW messages warn you that there is an issue. (2) relates to page set media recovery if the queue manager were to be shut down now and restarted. It does not know about when you last backed up your page sets, or which backup you might have to use if there was a page set failure. It normally moves to a more recent position in the log during checkpoint processing as changes held in the buffer pools are written to the page sets. In (3), the queue manager does know about CF structure backups taken either on this queue manager or on other queue managers in the queue sharing group. However, CF structure recovery requires a merge of log data from all queue managers in the queue sharing group which have interacted with the CF structure since the last backup. This means that the log data is identified by a log record sequence number, (or LRSN), which is timestamp based and so applicable across the entire queue sharing group rather than an RBA which would be different on different queue managers in the queue sharing group. It normally moves to a more recent position in the log as BACKUP CFSTRUCT commands are performed on either this or other queue managers in the queue sharing group.

## **Resetting the queue manager's log**

Use this topic to understand how to reset the queue manager's log.

You must not allow the queue manager log RBA to wrap around from the end of the log RBA range to 0, as this leads to a queue manager outage and all persistent data will become unrecoverable. The end of the log RBA is either a value of FFFFFFFFFF (if 6-byte RBAs as in use), or FFFFFFFFFFFFFFFF (if 8-byte RBAs are in use).

The queue manager issues messages [CSQI045I](#), [CSQI046E](#), [CSQI047E](#), [CSQJ031D](#), and [CSQJ032E](#) to indicate that the used log range is significant and that you should plan to take action to avoid an unplanned outage.

The queue manager terminates with reason code [00D10257](#) when the RBA value reaches FFF800000000 (if 6-byte log RBAs are in use) or FFFFFFFC0000000000 (if 8-byte log RBAs are in use).

If 6-byte log RBAs are in use, consider converting the queue manager to use 8-byte log RBAs rather than resetting the queue manager's log, following the process described in [“Implementing the larger log Relative Byte Address” on page 521](#). Converting a queue manager to use 8-byte log RBAs requires a shorter outage than resetting the log, and increases the period of time before you have to reset the log.

Message [CSQJ034I](#), issued during queue manager initialization, indicates the end of the log RBA range for the queue manager as configured, and can be used to determine whether 6-byte or 8-byte log RBAs are in use.

The procedure to follow to reset the queue manager's log is as follows:

1. Resolve any unresolved units of work. The number of unresolved units of work is displayed at queue manager startup in message CSQR005I as the INDOUBT count. At each checkpoint, and at queue manager shutdown, the queue manager automatically issues the command

**DISPLAY CONN(\*) TYPE(CONN) ALL WHERE(UOWSTATE EQ UNRESOLVED)** to provide information about unresolved units of work.

See [How in-doubt units of recovery are resolved](#) for information on resolving units of recovery. The ultimate recourse is to use the **RESOLVE INDOUBT** MQSC command to manually resolve indoubt units of recovery.

2. Shut down the queue manager cleanly.

You can use either **STOP QMGR** or **STOP QMGR MODE(FORCE)** as both these commands flush any changed pages from bufferpools to the page sets.

3. If a queue manager is part of a queue sharing group, take CFSTRUCT backups on other queue managers for all structures in the queue sharing group. This ensures that the most recent backups are not in this queue manager's log, and that this queue manager's log is not required for CFSTRUCT recovery.
4. Define new logs and BSDS using CSQJU003 (see [The change log inventory utility](#) for more information on using the change log inventory utility).
5. Run **CSQUTIL RESETPAGE** against all the page sets for this queue manager (see [Copying a page and resetting the log](#) for more information on using this function). Note that page set RBAs can be reset independently, so multiple concurrent jobs (for example, one per page set) can be submitted to reduce the elapsed time for this step.
6. Restart the queue manager

## Warning messages

When IBM MQ detects that the end of the log is approaching, it issues console messages in the following order, which indicate that a log reset should be planned. In this section the messages show 6-byte log RBA values. The same console messages are issued when IBM MQ is running in 8-byte log RBA mode but with different values; see [“Warning thresholds” on page 520](#) for the 8-byte log RBA thresholds.

1. When IBM MQ detects that the end of the log is approaching in the near future, (approximately 94% full) IBM MQ issues console message CSQI045I, as in the following example:

```
CSQI045I -CSQ7 CSQILCUR Log RBA has reached 0000F00000000000.  
Plan a log reset
```

2. IBM MQ issues the following CSQI046E error console message when the end of the log is near (approximately 97% full). This informs the IBM MQ administrator to take action soon.

```
CSQI046E -CSQ7 CSQILCUR Log RBA has reached 0000F80000000000.  
Perform a log reset
```

3. After the CSQI046E message is issued, at the next log switch, IBM MQ issues the following CSQJ032E console message with the word WARNING:

```
CSQJ032E -CSQ7 CSQJW307 WARNING - APPROACHING END OF  
THE LOG RBA RANGE OF 0000FFFFFFFF. CURRENT LOG RBA IS 0000F80000022000.
```

4. After the CSQI046E and CSQJ032E console messages are issued, IBM MQ issues one more error message, which does not require immediate IBM MQ administrator intervention. IBM MQ issues console message CSQI047E (when the log is approximately 99% full):

```
CSQI047E -CSQ7 CSQILCUR Log RBA has reached 0000FF0000000000.  
Stop queue manager and reset logs
```

5. When the log RBA reaches FF8000000000, IBM MQ increases the urgency of the situation and issues console message CSQJ032E with the word CRITICAL:

```
CSQJ032E -CSQ7 CSQJW009 CRITICAL - APPROACHING END OF THE LOG RBA RANGE OF 0000FFFFFFFFFFFF.
CURRENT LOG RBA IS 0000FFF7FFFFDFFF.
```

6. If the queue manager is started when the log RBA is almost at the maximum, the following CSQJ031D console message is issued. This stage requires the input of the IBM MQ administrator:.

```
CSQJ031D -CSQ7 CSQYSTR THE LOG RBA RANGE MUST BE RESET.
REPLY 'Y' TO CONTINUE STARTUP OR 'N' TO SHUTDOWN
```

7. IBM MQ startup remains suspended until a reply is given to message CSQJ031D.

The purpose of these messages is to give the IBM MQ administrator time to plan for a system outage to reset the logs. In an ideal configuration, there are at least two queue managers, possibly in a queue sharing group (QSG), sharing the workload. When one is down for maintenance the other can continue to receive work.

The severity of console messages that IBM MQ issues becomes greater as the RBA gets closer to the end. Ideally your IBM MQ administrator should plan to reset the log RBA when the first console message is seen.

If the warning and error console messages are ignored, IBM MQ terminates with reason code 5C6-00D10257 when the log RBA reaches FFF800000000, at which point IBM MQ determines that the available range is too small for the queue manager to continue. When this point is reached, the only option is to take an outage and either reset the log or extend the size of the log RBA.

**Note:** When the end of the log is reached it is not possible to resolve any in-flight units of work (UOW); these are lost during the log reset process. Enough of the RBA range should be left to start the queue manager and resolve any UOW. Because IBM MQ issues console messages several times to inform that the end of the log is approaching, a log reset should be planned.

The preferred option to avoid losing any in-flight UOW is to extend the log RBA to use 8 bytes. This means that a log RBA reset will not be necessary for a long period.

## Warning thresholds

The following table lists the thresholds, based on the length of the log RBA.

Console message	6-byte log RBA	8-byte log RBA
CSQI045I	0000F00000000000	FFFF800000000000
CSQI046E	0000F80000000000	FFFFC00000000000
CSQI047E	0000FF8000000000	FFFFFC0000000000
CSQJ032E	0000FF8000000000 0000FF8000000000	FFFFFC0000000000 FFFFFC0000000000
CSQJ031D	0000FF8000000000	FFFFFC0000000000

### Notes:

1. For message CSQJ032E, the first number applies to the WARNING text and the second number applies to the CRITICAL text in the console message.
2. Message CSQJ031D is issued at IBM MQ initialization only.

### Related concepts

[“Implementing the larger log Relative Byte Address” on page 521](#)

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.



## **Implementing the larger log Relative Byte Address**

Before IBM MQ for z/OS 8.0, IBM MQ for z/OS used a 6 byte log RBA to identify the location of data within the log. From IBM MQ for z/OS 8.0, the log RBA can be 8 bytes long, increasing the period of time before you have to reset the log.



**Attention:** You only have to carry out the following procedure to enable this feature if your queue managers were created before IBM MQ 9.3.0, as queue managers created at IBM MQ 9.3.0 and later already have this feature enabled.

See [Planning to increase the maximum addressable log range](#) for considerations when planning to enable 8 byte log RBA.

Perform these instructions, in the order shown, to enable 8 byte log RBA on a single IBM MQ for z/OS queue manager. For queue managers in a queue sharing group, perform the steps on each queue manager in turn.

1. Allocate new BSDS data sets with similar attributes to the current BSDS. You can tailor sample CSQ4BSDS and delete any irrelevant statement, or you can use your existing JCL, but change the BSDS name to something like ++HLQ++ .NEW .BSDS01.

### **Notes:**

- a. Check the attributes of your new BSDS before submitting the job to allocate the new BSDS. The only attribute that might change is the size of the BSDS.
  - b. The new BSDS contains more data than the current BSDS, therefore, you must ensure that the new data sets are allocated with sufficient available space. The sample JCL in `thlqual.SCSQPROC(CSQ4BSDS)` contains the recommended values when defining a new BSDS.
2. Shut down the queue manager cleanly.
  3. Run the [BSDS conversion utility \(CSQJUCNV\)](#) to convert the existing BSDS to the new BSDS data sets. This usually takes a few seconds to run.  
  
Your existing BSDS will not be changed during this process, and you can use that for the initialization of the queue manager in the case of an unsuccessful conversion.
  4. Rename the current BSDS to become the old BSDS, and the new BSDS to become the current BSDS, so that the new data sets are used when you next restart the queue manager. You can use the DFSMS Access Method Services ALTER command, for example:

```
ALTER '++HLQ++.BSDS01' NEWNAME('++HLQ++.OLD.BSDS01')
ALTER '++HLQ++.NEW.BSDS01' NEWNAME('++HLQ++.BSDS01')
```

Ensure that you also issue commands to rename both the data and index portions of the VSAM cluster.

5. Restart the queue manager. It should start in the same amount of time as it would have done when using 6 byte log RBA.

If the queue manager does not restart successfully due to a failure to access the converted BSDS, attempt to identify the cause of the failure, resolve the problem and retry the operation. If required, contact your IBM support center for assistance.

If necessary, the change can be backed out at this point by:

- a. Renaming the current BSDS to become the new BSDS.
  - b. Renaming the old BSDS to become the current BSDS.
  - c. Restarting the queue manager.
6. Once the queue manager has been successfully restarted with the converted BSDS, do not attempt to start the queue manager using the old BSDS.
  7. Message `CSQJ034I` is issued during queue manager initialization to indicate the end of the log RBA for the queue manager as configured. Confirm that the end of the log RBA range displayed is `FFFFFFFFFFFFFFFF`. This indicates that 8 byte log RBA is in use.

## Related tasks

[Planning to increase the maximum addressable log range](#)

## Related reference

[Larger log Relative Byte Address](#)

[The BSDS conversion utility \(CSQJUCNV\)](#)

## Managing the bootstrap data set (BSDS)

The bootstrap data set (BSDS) is used to reference log data sets, and log records. Use this topic to understand how you can examine, change, and recover the BSDS.

For more information, see [The bootstrap data set](#).

This topic describes the tasks involved in managing the bootstrap data set. It contains these sections:

- [“Finding out what the BSDS contains” on page 522](#)
- [“Changing the BSDS” on page 523](#)
- [“Recovering the BSDS” on page 527](#)

## Finding out what the BSDS contains

You can use the print log map utility (CSQJU004) to examine the contents of the BSDS.

The print log map utility (CSQJU004) is a batch utility that lists the information stored in the BSDS. For instructions on running it, see [The print log map utility](#).

The BSDS contains:

- [Time stamps](#)
- [Active log data set status](#)

### Time stamps in the BSDS

The output of the print log map utility shows the time stamps, which are used to record the date and time of various system events, that are stored in the BSDS.

The following time stamps are included in the header section of the report:

#### SYSTEM TIMESTAMP

Reflects the date and time the BSDS was last updated. The BSDS time stamp can be updated when:

- The queue manager starts.
- The write threshold is reached during log write activities. Depending on the number of output buffers you have specified and the system activity rate, the BSDS might be updated several times a second, or might not be updated for several seconds, minutes, or even hours. For details of the write threshold, see the WRTHRS parameter of the CSQ6LOGP macro in [Using CSQ6LOGP](#).
- IBM MQ drops into a single BSDS mode from its normal dual BSDS mode due to an error. This can occur when a request to get, insert, point to, update, or delete a BSDS record is unsuccessful. When this error occurs, IBM MQ updates the time stamp in the remaining BSDS to force a time stamp mismatch with the disabled BSDS.

#### UTILITY TIMESTAMP

The date and time the contents of the BSDS were altered by the change log inventory utility (CSQJU003).

The following time stamps are included in the active and archive log data sets portion of the report:

**Active log date**

The date the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

**Active log time**

The time the active log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done.

**Archive log date**

The date the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

**Archive log time**

The time the archive log entry was created in the BSDS, that is, when the CSQJU003 NEWLOG was done or the archive itself was done.

**Active log data set status**

The BSDS records the status of an active log data set as one of the following:

**NEW**

The data set has been defined but never used by IBM MQ, or the log was truncated to a point before the data set was first used. In either case, the data set starting and ending RBA values are reset to zero.

**REUSABLE**

Either the data set has been defined but never used by IBM MQ, or the data set has been offloaded. In the print log map output, the start RBA value for the last REUSABLE data set is equal to the start RBA value of the last archive log data set.

**NOT REUSABLE**

The data set contains records that have not been offloaded.

**STOPPED**

The offload processor encountered an error while reading a record, and that record could not be obtained from the other copy of the active log.

**TRUNCATED**

Either:

- An I/O error occurred, and IBM MQ has stopped writing to this data set. The active log data set is offloaded, beginning with the starting RBA and continuing up to the last valid record segment in the truncated active log data set. The RBA of the last valid record segment is lower than the ending RBA of the active log data set. Logging is switched to the next available active log data set, and continues uninterrupted.

or

- An ARCHIVE LOG function has been called, which has truncated the active log.

The status appears in the output from the print log map utility.

 **Changing the BSDS**

You do not have to take special steps to keep the BSDS updated with records of logging events because IBM MQ does that automatically.

However, you might want to change the BSDS if you do any of the following:

- Add more active log data sets.
- Copy active log data sets to newly allocated data sets, for example, when providing larger active log allocations.
- Move log data sets to other devices.
- Recover a damaged BSDS.
- Discard outdated archive log data sets.

You can change the BSDS by running the change log inventory utility (CSQJU003). Only run this utility when the queue manager is inactive, or you might get inconsistent results. The action of the utility is controlled by statements in the SYSIN data set. This section shows several examples. For complete instructions, see [The change log inventory utility](#).

You can copy an active log data set only when the queue manager is inactive because IBM MQ allocates the active log data sets as exclusive (DISP=OLD) at queue manager startup.

### *Changes for active logs*

Use this topic to understand how you can change the active logs using the BSDS.

You can add to, delete from, and record entries in the BSDS for active logs using the change log utility. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

See these sections for more information:

- [Adding record entries to the BSDS](#)
- [Deleting information about the active log data set from the BSDS](#)
- [Recording information about the log data set in the BSDS](#)
- [Increasing the size of the active log](#)
- [The use of CSQJUFMT](#)

#### **Adding record entries to the BSDS**

If an active log has been flagged as "stopped", it is not reused for logging; however, it continues to be used for reading. Use the access method services to define new active log data sets, then use the change log inventory utility to register the new data sets in the BSDS. For example, use:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY1
NEWLOG DSNAMES=MQM111.LOGCOPY2.DS10,COPY2
```

If you are copying the contents of an old active log data set to the new one, you can also give the RBA range and the starting and ending time stamps on the NEWLOG function.

#### **Deleting information about the active log data set from the BSDS**

To delete information about an active log data set from the BSDS, you could use:

```
DELETE DSNAMES=MQM111.LOGCOPY1.DS99
DELETE DSNAMES=MQM111.LOGCOPY2.DS99
```

#### **Recording information about the log data set in the BSDS**

To record information about an existing active log data set in the BSDS, use:

```
NEWLOG DSNAMES=MQM111.LOGCOPY1.DS10,COPY2,STARTIME=19930212205198,
ENDTIME=19930412205200,STARTRBA=6400,ENDRBA=94FF
```

You might need to insert a record containing this type of information in the BSDS because:

- The entry for the data set has been deleted, but is needed again.
- You are copying the contents of one active log data set to another data set.
- You are recovering the BSDS from a backup copy.

## Increasing the size of the active log

There are two methods of achieving this process.

1. When the queue manager is active:
  - a. Define new larger log data sets using JCL.
  - b. Add the new log data sets to the active queue manager using the MQSC DEFINE LOG command.
  - c. Use the MQSC ARCHIVE LOG command to move the current active log, to be a new larger log.
  - d. Wait for the archive of the smaller active log data set to complete.
  - e. Shut down the queue manager, using the CSQJU003 utility to remove the old small active logs.
  - f. Restart the queue manager.
2. When the queue manager is inactive:
  - a. Stop the queue manager. This step is required because IBM MQ allocates all active log data sets for its exclusive use when it is active.
  - b. Use Access Method Services ALTER with the NEWNAME option to rename your active log data sets.
  - c. Use Access Method Services DEFINE to define larger active log data sets.

By reusing the old data set names, you do not have to run the change log inventory utility to establish new names in the BSDSs. The old data set names and the correct RBA ranges are already in the BSDSs.
  - d. Use Access Method Services REPRO to copy the old (renamed) data sets into their appropriate new data sets.

**Note:** This step can take a long time, so your enterprise could be out of action for this period.
  - e. Start the queue manager.

If all your log data sets are the same size, your system will be operationally more consistent and efficient. If the log data sets are not the same size, it is more difficult to track your system's logs, and so space can be wasted.

### The use of CSQJUFMT

Do not run a CSQJUFMT format when increasing the size of an active log.

If you run CSQJUFMT (in order to provide a performance advantage the first time the queue manager writes to the new active log) you receive messages:

```
IEC070I 203-204,XS95GTLX,REPRO02,OUTPUT,B857,SPMG02, 358
IEC070I MG.W.MG4E.LOGCOPY1.DS02,MG.W.MG4E.LOGCOPY1.DS02.DATA,
IDC3302I ACTION ERROR ON MG.W.MG4E.LOGCOPY1.DS02
IDC3351I ** VSAM I/O RETURN CODE IS 28 - RPLFDBWD = X'2908001C'
IDC31467I MAXIMUM ERROR LIMIT REACHED.

IDC0005I NUMBER OF RECORDS PROCESSED WAS 0
```

In addition, if you use the Access Method Services REPRO, ensure that you define a new empty log.

If you use REPRO to copy the old (renamed) data set into its respective new data set, the default is NOREPLACE.

This means that REPRO does not replace a record that is already on the designated data set. When formatting is done on the data set, the RBA value is reset. The net result is a data set that is not empty after formatting.

### *Changes for archive logs*

Use this topic to understand how to change the archive logs.

You can add to, delete from, and change the password of, entries in the BSDS for archive logs. Examples only are shown here; replace the data set names shown with the ones you want to use. For more details of the utility, see [The change log inventory utility](#).

- [Adding an archive log](#)
- [Deleting an archive log](#)
- [Changing the password of an archive log](#)

### Adding an archive log

When the recovery of an object depends on reading an existing archive log data set, the BSDS must contain information about that data set so that IBM MQ can find it. To register information about an existing archive log data set in the BSDS, use:

```
NEWLOG DSNNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04,  
UNIT=TAPE,STARTRBA=3A190000,ENDRBA=3A1F0FFF,CATALOG=NO
```

### Deleting an archive log

To delete an entire archive log data set on one or more volumes, use:

```
DELETE DSNNAME=CSQARC1.ARCHLOG1.E00021.T2205197.A0000015,COPY1VOL=CSQV04
```

### Changing the password of an archive log

If you change the password of an existing archive log data set, you must also change the information in the BSDS.

1. List the BSDS, using the print log map utility.
2. Delete the entry for the archive log data set with the changed password, using the DELETE function of the CSQJU003 utility (see topic [The change log inventory utility](#)).
3. Name the data set as for a new archive log data set. Use the NEWLOG function of the CSQJU003 utility (see topic [The change log inventory utility](#)), and give the new password, the starting and ending RBAs, and the volume serial numbers (which can be found in the print log map utility output, see [The print log map utility](#)).


To change the password for new archive log data sets, use:

```
ARCHIVE PASSWORD= password
```

To stop placing passwords on new archive log data sets, use:

```
ARCHIVE NOPASSWD
```

**Note:** Only use the ARCHIVE utility function if you do not have an external security manager.

 **Changing the high-level qualifier (HLQ) for the logs and BSDS**

Use this topic to understand the procedure required to change the high-level qualifier (HLQ).

## Before you begin

You must end the queue manager normally before copying any of the logs or data sets to the new data sets. This is to ensure that the data is consistent and no recovery is needed during restart.

## About this task

This task provides information about how to change the HLQ for the logs and BSDS. To do this, follow these steps:

## Procedure

1. Run the log print utility CSQJU004 to record the log data set information. This information is needed later.
2. You can either:
  - a) run DSS backup and restore with rename on the log and BSDS data sets to be renamed, or
  - b) use AMS DEFINE and REPRO to create the HLQ data sets and copy the data from the old data sets.
3. Modify the MSTR and CHIN procedures to point to the new data sets.
4. Delete the old log information in the new copy of the BSDS using CSQJU003.
5. Define the new log data sets to the new BSDS using the NEWLOG function of CSQJU003.  
Keep all information about each log the same, apart from the HLQ.
6. The new BSDS should reflect the same information that was recorded for the old logs in the old BSDS.  
The HLQ should be the only thing that has changed.

## What to do next

Compare the CSQJU004 output for the old and new BSDS to ensure that they look EXACTLY the same (except for the HLQs) before starting the queue manager.

**Note:** Care must be taken when performing these operations. Incorrect actions might lead to unrecoverable situations. Check the PRINT LOG MAP UTILITY output and make sure that all the information needed for recovery or restart has been included.

## **Recovering the BSDS**

If IBM MQ is operating in dual BSDS mode and one BSDS becomes damaged, forcing IBM MQ into single BSDS mode, IBM MQ continues to operate without a problem (until the next restart).

To return the environment to dual BSDS mode:

1. Use Access Method Services to rename or delete the damaged BSDS and to define a new BSDS with the same name as the damaged BSDS. Example control statements can be found in job CSQ4BREC in thlqual.SCSQPROC.
2. Issue the IBM MQ command RECOVER BSDS to make a copy of the valid BSDS in the newly allocated data set and to reinstate dual BSDS mode.

If IBM MQ is operating in single BSDS mode and the BSDS is damaged, or if IBM MQ is operating in dual BSDS mode and both BSDSs are damaged, the queue manager stops and does not restart until the BSDS data sets are repaired. In this case:

1. Locate the BSDS associated with the most recent archive log data set. The data set name of the most recent archive log appears on the job log in the last occurrence of message CSQJ003I, which indicates that offload processing has been completed successfully. In preparation for the rest of this procedure, it is a good practice to keep a log of all successful archives noted by that message:
  - If archive logs are on DASD, the BSDS is allocated on any available DASD. The BSDS name is like the corresponding archive log data set name; change only the first letter of the last qualifier, from A to B, as in this example:

**Archive log name**

CSQ.ARCHLOG1. **A** 0000001

**BSDS copy name**

CSQ.ARCHLOG1. **B** 0000001

- If archive logs are on tape, the BSDS is the first data set of the first archive log volume. The BSDS is not repeated on later volumes.
- 2. If the most recent archive log data set has no copy of the BSDS (for example, because an error occurred when offloading it), locate an earlier copy of the BSDS from earlier offload processing.
- 3. Rename *damaged* BSDSs using the Access Method Services ALTER command with the NEWNAME option. If you want to delete a damaged BSDS, use the Access Method Services DELETE command. For each damaged BSDS, use Access Method Services to define a new BSDS as a replacement data set. Job CSQ4BREC in thlqual.SCSQPROC contains Access Method Services control statements to define a new BSDS.
- 4. Use the Access Method Services REPRO command to copy the BSDS from the archive log to one of the replacement BSDSs you defined in step “3” on page 528. Do not copy any data to the second replacement BSDS, you do that in step “5” on page 529.

- a. Print the contents of the replacement BSDS.

Use the print log map utility (CSQJU004) to print the contents of the replacement BSDS. This enables you to review the contents of the replacement BSDS before continuing your recovery work.

- b. Update the archive log data set inventory in the replacement BSDS.

Examine the output from the print log map utility and check that the replacement BSDS does not contain a record of the archive log from which the BSDS was copied. If the replacement BSDS is an old copy, its inventory might not contain all archive log data sets that were created more recently. The BSDS inventory of the archive log data sets must be updated to reflect the current subsystem inventory.

Use the change log inventory utility (CSQJU003) NEWLOG statement to update the replacement BSDS, adding a record of the archive log from which the BSDS was copied. If the archive log data set is password-protected, use the PASSWORD option of the NEWLOG function. Also, if the archive log data set is cataloged, ensure that the CATALOG option of the NEWLOG function is properly set to CATALOG=YES. Use the NEWLOG statement to add any additional archive log data sets that were created later than the BSDS copy.

- c. Update passwords in the replacement BSDS.

The BSDS contains passwords for the archive log data sets and for the active log data sets. To ensure that the passwords in the replacement BSDS reflect the current passwords used by your installation, use the change log inventory ARCHIVE utility function with the PASSWORD option.

- d. Update the active log data set inventory in the replacement BSDS.

In unusual circumstances, your installation might have added, deleted, or renamed active log data sets since the BSDS was copied. In this case, the replacement BSDS does not reflect the actual number or names of the active log data sets your installation currently has in use.

If you need to delete an active log data set from the replacement BSDS log inventory, use the change log inventory utility DELETE function.

If you need to add an active log data set to the replacement BSDS log inventory, use the change log inventory utility NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

If you need to rename an active log data set in the replacement BSDS log inventory, use the change log inventory utility DELETE function, followed by the NEWLOG function. Ensure that the RBA range is specified correctly on the NEWLOG function. If the active log data set is password-protected, use the PASSWORD option.

- e. Update the active log RBA ranges in the replacement BSDS.



Later, when the queue manager restarts, it compares the RBAs of the active log data sets listed in the BSDS with the RBAs found in the actual active log data sets. If the RBAs do not agree, the queue manager does not restart. The problem is magnified when an old copy of the BSDS is used. To solve this problem, use the change log inventory utility (CSQJU003) to adjust the RBAs found in the BSDS using the RBAs in the actual active log data sets. You do this by:

- Using the print log records utility (CSQ1LOGP) to print a summary report of the active log data set. This shows the starting and ending RBAs.
- Comparing the actual RBA ranges with the RBA ranges you have just printed, when the RBAs of all active log data sets are known.

If the RBA ranges are equal for all active log data sets, you can proceed to the next recovery step without any additional work.

If the RBA ranges are not equal, adjust the values in the BSDS to reflect the actual values. For each active log data set that needs to have the RBA range adjusted, use the change log inventory utility DELETE function to delete the active log data set from the inventory in the replacement BSDS. Then use the NEWLOG function to redefine the active log data set to the BSDS. If the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function.

- f. If only two active log data sets are specified for each copy of the active log, IBM MQ can have difficulty during queue manager restart. The problem can arise when one of the active log data sets is full and has not been offloaded, while the second active log data set is close to filling. In this case, add a new active log data set for each copy of the active log and define each new active log data set in the replacement BSDS log inventory.

Use the Access Method Services DEFINE command to define a new active log data set for each copy of the active log and use the change log inventory utility NEWLOG function to define the new active log data sets in the replacement BSDS. You do not need to specify the RBA ranges on the NEWLOG statement. However, if the active log data sets are password-protected, use the PASSWORD option of the NEWLOG function. Example control statements to accomplish this task can be found in job CSQ4LREC in thlqual.SCSQPROC.

5. Copy the updated BSDS to the second new BSDS data set. The BSDSs are now identical.

Use the print log map utility (CSQJU004) to print the contents of the second replacement BSDS at this point.

6. See [Active log problems](#) for information about what to do if you have lost your current active log data set.
7. Restart the queue manager using the newly constructed BSDS. IBM MQ determines the current RBA and what active logs need to be archived.

## Managing page sets

Use this topic to understand how to manage the page sets associated with a queue manager.

This topic describes how to add, copy, and generally manage the page sets associated with a queue manager. It contains these sections:

- [“How to change the high-level qualifier \(HLQ\) for the page sets” on page 530](#)
- [“How to add a page set to a queue manager” on page 530](#)
- [“What to do when one of your page sets becomes full” on page 530](#)
- [“How to balance loads on page sets” on page 531](#)
- [How to increase the size of a page set](#)
- [“How to reduce a page set” on page 534](#)
- [“How to reintroduce a page set” on page 535](#)

- [“How to back up and recover page sets” on page 536](#)
- [“How to delete page sets” on page 539](#)
- [“How to back up and restore queues using CSQUTIL” on page 539](#)

See [Page sets](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

## How to change the high-level qualifier (HLQ) for the page sets

This task gives information on how to change the HLQ for the page sets. To perform this task, do the following:

1. Define the new HLQ page sets.
2. If the size allocation is the same as the old page sets, copy the existing page set using REPRO to the empty new HLQ page sets.
3. If you are increasing the size of the page sets, use the FORMAT function of CSQUTIL to format the destination pages, and then the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set.

For more information, see [Formatting page sets \(FORMAT\)](#), and [Expanding a page set \(COPYPAGE\)](#).

4. Change the CSQP00xx DD statement in the queue manager procedure to point to the new HLQ page sets.

Restart the queue manager and verify the changes to the page sets.

## How to add a page set to a queue manager

This description assumes that you have a queue manager that is already running. You might need to add a page set if, for example, your queue manager has to cope with new applications using new queues.

To add a new page set, use the following procedure:

1. Define and format the new page set. You can use the sample JCL in thlqual.SCSQPROC(CSQ4PAGE) as a basis. For more information, see [Formatting page sets \(FORMAT\)](#).

Take care not to format any page sets that are in use, unless this is what you intend. If so, use the FORCE option of the FORMAT utility function.

2. Use the DEFINE PSID command with the DSN option to associate the page set with a buffer pool.
3. Add the appropriate storage class definitions for your page set by issuing DEFINE STGCLASS commands.
4. Optionally, to document how your queue manager is configured:
  - a. Add the new page set to the started task procedure for your queue manager.
  - b. Add a definition for the new page set to your CSQINP1 initialization data set.
  - c. Add a definition for the new storage class to your CSQ4INZR initialization data set member.

For details of the DEFINE PSID and DEFINE STGCLASS commands, see [DEFINE PSID](#) and [DEFINE STGCLASS](#).

## What to do when one of your page sets becomes full

You can find out about the utilization of page sets by using the IBM MQ command DISPLAY USAGE. For example, the command:

```
DISPLAY USAGE PSID(03)
```

displays the current state of the page set 03. This tells you how many free pages this page set has.

If you have defined secondary extents for your page sets, they are dynamically expanded each time they fill up. Eventually, all secondary extents are used, or no further disk space is available. If this happens, an application receives the return code MQRC\_STORAGE\_MEDIUM\_FULL.

If an application receives a return code of MQRC\_STORAGE\_MEDIUM\_FULL from an MQI call, this is a clear indication that there is not enough space remaining on the page set. If the problem persists or is likely to recur, you must do something to solve it.

You can approach this problem in a number of ways:

- Balance the load between page sets by moving queues from one page set to another.
- Expand the page set. See [“How to increase the size of a page set”](#) on page 533 for instructions.
- Redefine the page set so that it can expand beyond 4 GB to a maximum size of 64 GB. See [Defining a page set to be larger than 4 GB](#) for instructions.

## How to balance loads on page sets

Load balancing on page sets means moving the messages associated with one or more queues from one page set to another, less used, page set. Use this technique if it is not practical to expand the page set.

To identify which queues are using a page set, use the appropriate IBM MQ commands. For example, to find out which queues are mapped to page set 02, first, find out which storage classes map to page set 02, by using the command:

```
DISPLAY STGCLASS(*) PSID(02)
```

Then use the following command to find out which queues use which storage class:

```
DISPLAY QUEUE(*) TYPE(QLOCAL) STGCLASS
```

## Moving a non-shared queue

To move queues and their messages from one page set to another, use the MQSC MOVE QLOCAL command (described in MOVE QLOCAL). When you have identified the queue or queues that you want to move to a new page set, follow this procedure for each of these queues:

1. Ensure that the queue you want to move is not in use by any applications (that is, IPPROCS and OPPROCS values from the DISPLAY QSTATUS command are zero) and that it has no uncommitted messages (the UNCOM value from the DISPLAY QSTATUS command is NO).

**Note:** The only way to ensure that this state continues is to change the security authorization of the queue temporarily. See [Profiles for queue security](#) for more information.

If you cannot do this, later stages in this procedure might fail if applications start to use the queue despite precautionary steps such as setting PUT(DISABLED). However, messages can never be lost by this procedure.

2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved, using the command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
```

**Note:** If this temporary queue already exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Define a new storage class that maps to the required page set, for example:

```
DEFINE STGCLASS(NEW) PSID(nn)
```

Add the new storage class definition to the CSQINP2 data sets ready for the next queue manager restart.

7. Redefine the queue that you are moving, by changing the storage class attribute:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 531.

8. Move the messages back to the new queue, using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

9. The queue created in step “3” on page 531 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

10. If the queue being moved was defined in the CSQINP2 data sets, change the STGCLASS attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 29 on page 533 shows an extract from a load balancing job.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
//      DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_To_Move) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_To_Move)
DEFINE STGCLASS(NEW) PSID(2)
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) STGCLASS(NEW)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_To_Move)
DELETE QL(TEMP_QUEUE)
/*

```

Figure 29. Extract from a load balancing job for a page set

## How to increase the size of a page set

You can initially allocate a page set larger than 4 GB, See [Defining a page set to be larger than 4 GB](#)

A page set can be defined to be automatically expanded as it becomes full by specifying EXPAND(SYSTEM) or EXPAND(USER). If your page set was defined with EXPAND(NONE), you can expand it in either of two ways:

- Alter its definition to allow automatic expansion. See [Altering a page set to allow automatic expansion](#)
- Create a new, larger page set and copy the messages from the old page set to the new one. See [Moving messages to a new, larger page set](#)

### Defining a page set to be larger than 4 GB

IBM MQ can use a page set up to 64 GB in size, provided the data set is defined with 'extended addressability' to VSAM. Extended addressability is an attribute which is conferred by an SMS data class.

**Note:** Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and, from z/OS V1.12, an archive log dataset can also reside in the EAS.

In the example shown in the following sample JCL, the management class 'EXTENDED' is defined to SMS with 'Extended addressability'. If your existing page set is not currently defined as having extended addressability, use the following method to migrate to an extended addressability format data set.

1. Stop the queue manager.
2. Use Access Method Services to rename the existing page set.
3. Define a destination page set, the same size as the existing page set, but with DATACLAS(EXTENDED).

**Note:** Extended-format data sets must be SMS managed. These are the mechanisms for requesting extended format for VSAM data sets:

- Using a data class that has a DSNTYPE value of EXT and the subparameter R or P to indicate required or preferred.
- Coding DSNTYPE=EXTREQ (extended format is required) or DSNTYPE=EXTPREF (extended format is preferred) on the DD statement.

- Coding the LIKE= parameter on the DD statement to refer to an existing extended format data set.

For more information, see [Restrictions on Defining Extended-Format Data Sets](#).

4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager.
6. Alter the page set to use system expansion, to allow it to continue growing beyond its current allocation.

The following JCL shows example Access Method Services commands:

```
//S1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
ALTER 'VICY.CSQ1.PAGE01' -
NEWNAME('VICY.CSQ1.PAGE01.OLD')
ALTER 'VICY.CSQ1.PAGE01.DATA' -
NEWNAME('VICY.CSQ1.PAGE01.DATA.OLD')
DEFINE CLUSTER (NAME('VICY.CSQ1.PAGE01') -
MODEL('VICY.CSQ1.PAGE01.OLD') -
DATACLAS(EXTENDED))
/*
```

### Altering a page set to allow automatic expansion

Use the ALTER PSID command with the EXPAND(USER) or EXPAND(SYSTEM) options. See [ALTER PSID](#) and [Expanding a page set \(COPYPAGE\)](#) for general information on expanding page sets.

### Moving messages to a new, larger page set

This technique involves stopping and restarting the queue manager. This deletes any nonpersistent messages that are not on shared queues at restart time. If you have nonpersistent messages that you do not want to be deleted, use load balancing instead. For more details, see [“How to balance loads on page sets” on page 531](#). In this description, the page set that you want to expand is referred to as the *source* page set; the new, larger page set is referred to as the *destination* page set.

Follow these steps:

1. Stop the queue manager.
2. Define the destination page set, ensuring that it is larger than the source page set, with a larger secondary extent value.
3. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets \(FORMAT\)](#) for more details.
4. Use the COPYPAGE function of CSQUTIL to copy all the messages from the source page set to the destination page set. See [Expanding a page set \(COPYPAGE\)](#) for more details.
5. Restart the queue manager using the destination page set by doing one of the following:
  - Change the queue manager started task procedure to reference the destination page set.
  - Use Access Method Services to delete the source page set and then rename the destination page set, giving it the same name as that of the source page set.

#### Attention:

Before you delete any IBM MQ page set, be sure that you have made the required backup copies.

### How to reduce a page set

Prevent all users, other than the IBM MQ administrator, from using the queue manager. For example; by changing the access security settings.

If you have a large page set that is mostly empty (as shown by the DISPLAY USAGE command), you might want to reduce its size. The procedure to do this involves using the COPY, FORMAT, and LOAD functions of CSQUTIL (see [IBM MQ utility program](#)). This procedure does not work for page set zero (0), as it is not practical to reduce the size of this page set; the only way to do so is by reinitializing your queue manager (see [“Reinitializing a queue manager” on page 557](#)). The prerequisite of this procedure is to try and remove all users from the system so that all UOWs are complete and the page sets are consistent.

1. Use the STOP QMGR command with the QUIESCE or FORCE attribute to stop the queue manager.
2. Run the SCOPY function of CSQUTIL with the PSID option, to copy all message data from the large page set and save them in a sequential data set.
3. Define a new smaller page set data set to replace the large page set.
4. Run the FORMAT TYPE(NEW) function of CSQUTIL against the page set that you created in step [“3” on page 535](#).
5. Restart the queue manager using the page set created in step [“3” on page 535](#).
6. Run the LOAD function of CSQUTIL to load back all the messages saved during step [“2” on page 535](#).
7. Allow all users access to the queue manager.
8. Delete the old large page set.

## How to reintroduce a page set

In certain scenarios it is useful to be able to bring an old page set online again to the queue manager. Unless specific action is taken, when the old page set is brought online the queue manager will recognize that the page set recovery RBA stored in the page set itself and in the checkpoint records is old, and will therefore automatically start media recovery of the page set to bring it up to date.

Such media recovery can only be performed at queue manager restart, and is likely to take a considerable length of time, especially if archive logs held on tape must be read. However, normally in this circumstance, the page set has been offline for the intervening period and so the log contains no information pertinent to the page set recovery.

The following three choices are available:

### **Allow full media recovery to be performed.**

1. Stop the queue manager.
2. Ensure definitions are available for the page set in both the started task procedure for the queue manager and in the CSQINP1 initialization data set.
3. Restart the queue manager.

### **Allow any messages on the page set to be destroyed.**

This choice is useful where a page set has been offline for a long time (some months, for example) and it has now been decided to reuse it for a different purpose.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(NEW) option.
2. Add definitions for the page set to both the started task procedure for the queue manager and the CSQINP1 initialization data set.
3. Restart the queue manager.

Using the TYPE(NEW) option for formatting clears the current contents of the page set and tells the queue manager to ignore any historical information in the checkpoint about the page set.

### **Bring the page set online avoiding the media recovery process.**

Use this technique only if you are sure that the page set has been offline since a clean shutdown of the queue manager. This choice is most appropriate where the page set has been offline for a short period, typically due to operational issues such as a backup running while the queue manager is being started.

1. Format the page set using the FORMAT function of CSQUTIL with the TYPE(REPLACE) option.

2. Either add the page set back into the queue manager dynamically using the DEFINE PSID command with the DSN option or allow it to be added at a queue manager restart.

Using the TYPE(REPLACE) option for formatting checks that the page set was cleanly closed by the queue manager, and marks it so that media recovery will not be performed. No other changes are made to the contents of the page set.

## **How to back up and recover page sets**

There are different mechanisms available for back up and recovery. Use this topic to understand these mechanisms.

This section describes the following topics:

- [“Creating a point of recovery for non-shared resources” on page 536](#)
- [“Backing up page sets” on page 537](#)
- [“Recovering page sets” on page 538](#)
- [How to delete page sets](#)

For information about how to create a point of recovery for shared resources, see [“Recovering shared queues” on page 544](#).

### **Creating a point of recovery for non-shared resources**

IBM MQ can recover objects and non-shared persistent messages to their current state if both:

1. Copies of page sets from an earlier point exist.
2. All the IBM MQ logs are available to perform recovery from that point.

These represent a point of recovery for non-shared resources.

Both objects and messages are held on page sets. Multiple objects and messages from different queues can exist on the same page set. For recovery purposes, objects and messages cannot be backed up in isolation, so a page set must be backed up as a whole to ensure the correct recovery of the data.

The IBM MQ recovery log contains a record of all persistent messages and changes made to objects. If IBM MQ fails (for example, due to an I/O error on a page set), you can recover the page set by restoring the backup copy and restarting the queue manager. IBM MQ applies the log changes to the page set from the point of the backup copy.

There are two ways of creating a point of recovery:

#### **Full backup**

Stop the queue manager, which forces all updates on to the page sets.

This allows you to restart from the point of recovery, using only the backed up page set data sets and the logs from that point on.

#### **Fuzzy backup**

Take *fuzzy* backup copies of the page sets without stopping the queue manager.

If you use this method, and your associated logs later become damaged or lost, you cannot use the fuzzy page set backup copies to recover. This is because the fuzzy page set backup copies contain an inconsistent view of the state of the queue manager and are dependent on the logs being available. If the logs are not available, you need to return to the last set of backup page set copies taken while the subsystem was inactive ([Method 1](#)) and accept the loss of data from that time.

#### **Method 1: Full backup**

This method involves shutting the queue manager down. This forces all updates on to the page sets so that the page sets are in a consistent state.



1. Stop all the IBM MQ applications that are using the queue manager (allowing them to complete first). This can be done by changing the access security or queue settings, for example.
2. When all activity has completed, display and resolve any in-doubt units of recovery. (Use the commands `DISPLAY CONN` and `RESOLVE INDOUBT`, as described in [DISPLAY CONN](#) and [RESOLVE INDOUBT](#).)  
This brings the page sets to a consistent state; if you do not do this, your page sets might not be consistent, and you are effectively doing a fuzzy backup.
3. Issue the `ARCHIVE LOG` command to ensure that the latest log data is written out to the log data sets.
4. Issue the `STOP QMGR MODE(QUIESCE)` command. Record the lowest RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information). You should keep the log data sets starting from the one indicated by the RBA value up to the current log data set.
5. Take backup copies of all the queue manager page sets (see [“Backing up page sets”](#) on page [537](#)).

## Method 2: Fuzzy backup

This method does not involve shutting the queue manager down. Therefore, updates might be in virtual storage buffers during the backup process. This means that the page sets are not in a consistent state, and can only be used for recovery with the logs.

1. Issue the `DISPLAY USAGE TYPE(ALL)` command, and record the RBA value in the `CSQI024I` or `CSQI025I` messages (see [CSQI024I](#) and [CSQI025I](#) for more information).
2. Take backup copies of the page sets (see [“Backing up page sets”](#) on page [537](#)).
3. Issue the `ARCHIVE LOG` command, to ensure that the latest log data is written out to the log data sets. To restart from the point of recovery, you must keep the log data sets starting from the log data set indicated by the RBA value up to the current log data set.

## Backing up page sets

To recover a page set, IBM MQ needs to know how far back in the log to go. IBM MQ maintains a log RBA number in page zero of each page set, called the *recovery log sequence number* (LSN). This number is the starting RBA in the log from which IBM MQ can recover the page set. When you back up a page set, this number is also copied.

If the copy is later used to recover the page set, IBM MQ must have access to all the log records from this RBA value to the current RBA. That means you must keep enough of the log records to enable IBM MQ to recover from the oldest backup copy of a page set you intend to keep.

Use `ADRDSU COPY` function to copy the page sets.

For more information, see the [COPY DATASET Command Syntax for Logical Data Set](#) documentation .

For example:

```
//STEP2 EXEC PGM=ADRDSU,REGION=6M
//SYSPRINT DD SYSOUT=H
//SYSIN DD *
COPY -
DATASET(INCLUDE(SCENDATA.MQPA.PAGESET.*)) -
RENAMEU(SCENDATA.MQPA.PAGESET.** ,SCENDATA.MQPA.BACKUP1.** ) -
SPHERE -
REPUNC -
FASTREPLICATION(PREF ) -
CANCELERROR -
TOL(ENQF)
/*
//
```

If you copy the page set while the queue manager is running you must use a copy utility that copies page zero of the page set first. If you do not do this you could corrupt the data in your page set.

If the process of dynamically expanding a page set is interrupted, for example by power to the system being lost, you can still use ADRDSSU to take a backup of a page set.

If you perform an Access Method Services IDCAMS LISTCAT ENT('page set data set name') ALLOC, you will see that the HI-ALLOC-RBA is higher than the HI-USED-RBA.

The next time this page set fills up it is extended again, if possible, and the pages between the high used RBA and the highest allocated RBA are used, along with another new extent.

## Backing up your object definitions

You should also back up copies of your object definitions. To do this, use the MAKEDEF feature of the CSQUTIL COMMAND function (described in [Issuing commands to IBM MQ \(COMMAND\)](#)).

Back up your object definitions whenever you take a backup copy of your queue manager, and keep the most current version.

## Recovering page sets

If the queue manager has terminated due to a failure, the queue manager can normally be restarted with all recovery being performed during restart. However, such recovery is not possible if any of your page sets or log data sets are not available. The extent to which you can now recover depends on the availability of backup copies of page sets and log data sets.

To restart from a point of recovery you must have:

- A backup copy of the page set that is to be recovered.
- If you used the "fuzzy" backup process described in ["Method 2: Fuzzy backup" on page 537](#), the log data set that included the recorded RBA value, the log data set that was made by the ARCHIVE LOG command, and all the log data sets between these.
- If you used full backup, but you do not have the log data sets following that made by the ARCHIVE LOG command, you do **not** need to run the FORMAT TYPE(REPLACE) function of the CSQUTIL utility against all your page sets.

To recover a page set to its current state, you must also have all the log data sets and records since the ARCHIVE LOG command.

There are two methods for recovering a page set. To use either method, the queue manager must be stopped.

### Simple recovery

This is the simpler method, and is appropriate for most recovery situations.

1. Delete the page set you want to restore from backup.
2. Use the ADRDSSU COPY function to recover your page set from the backup copy..

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Restart the queue manager.
4. When the queue manager has restarted successfully, you can restart your applications
5. Reinstate your normal backup procedures for the restored page.

### Advanced recovery

This method provides performance advantages if you have a large page set to recover, or if there has been much activity on the page set since the last backup copy was taken. However, it requires more manual intervention than the simple method, which might increase the risk of error and the time taken to perform the recovery.

1. Delete and redefine the page set you want to restore from backup.
2. Use ADRDSSU to copy the backup copy of the page set into the new page set. Define your new page set with a secondary extent value so that it can be expanded dynamically.

Alternatively, you can rename your backup copy to the original name, or change the CSQP00xx DD statement in your queue manager procedure to point to your backup page set. However, if you then lose or corrupt the page set, you will no longer have a backup copy to restore from.

3. Change the CSQINP1 definitions for your queue manager to make the buffer pool associated with the page set being recovered as large as possible. By making the buffer pool large, you might be able to keep all the changed pages resident in the buffer pool and reduce the amount of I/O to the page set.
4. Restart the queue manager.
5. When the queue manager has restarted successfully, stop it (using quiesce) and then restart it using the normal buffer pool definition for that page set. After this second restart completes successfully, you can restart your applications
6. Reinststate your normal backup procedures for the restored page.

### What happens when the queue manager is restarted

When the queue manager is restarted, it applies all changes made to the page set that are registered in the log, beginning at the restart point for the page set. IBM MQ can recover multiple page sets in this way. The page set is dynamically expanded, if required, during media recovery.

During restart, IBM MQ determines the log RBA to start from by taking the lowest value from the following:

- Recovery LSN from the checkpoint log record for each page set.
- Recovery LSN from page zero in each page set.
- The RBA of the oldest incomplete unit of recovery in the system at the time the backup was taken.

All object definitions are stored on page set zero. Messages can be stored on any available page set.

**Note:** The queue manager cannot restart if page set zero is not available.

### How to delete page sets

You delete a page set by using the DELETE PSID command; see [DELETE PSID](#) for details of this command.

You cannot delete a page set that is still referenced by any storage class. Use DISPLAY STGCLASS to find out which storage classes reference a page set.

The data set is deallocated from IBM MQ but is not deleted. It remains available for future use, or can be deleted using z/OS facilities.

Remove the page set from the started task procedure for your queue manager.

Remove the definition of the page set from your CSQINP1 initialization data set.

### **How to back up and restore queues using CSQUTIL**

Use this topic as a reference for further information about back up and restore using CSQUTIL.

You can use the CSQUTIL utility functions for backing up and restoring queues. To back up a queue, use the COPY or SCOPY function to copy the messages from a queue onto a data set. To restore the queue, use the complementary function LOAD or SLOAD. For more information, see [IBM MQ utility program](#).

### **Managing buffer pools**

Use this topic if you want to change or delete your buffer pools.

This topic describes how to alter and delete buffer pools. It contains these sections:

- [“How to change the number of buffers in a buffer pool” on page 540](#)
- [“How to delete a buffer pool” on page 540](#)

Buffer pools are defined during queue manager initialization, using [DEFINE BUFFPOOL](#) commands issued from the initialization input data set CSQINP1. Their attributes can be altered in response to business requirements while the queue manager is running, using the processes detailed in this topic. The queue manager records the current buffer pool attributes in checkpoint log records. These are automatically restored on subsequent queue manager restart, unless the buffer pool definition in CSQINP1 includes the REPLACE attribute.

Use the [DISPLAY USAGE](#) command to display the current buffer attributes.

You can also define buffer pools dynamically using the [DEFINE PSID](#) command with the DSN option.

If you change buffer pools dynamically, you should also update their definitions in the initialization data set CSQINP1.

See [Planification sur z/OS](#) for a description of page sets, storage classes, buffers, and buffer pools, and some of the performance considerations that apply.

**Note:** Buffer pools use significant storage. When you increase the size of a buffer pool or define a new buffer pool ensure that sufficient storage is available. For more information, see [Address space storage](#).

## How to change the number of buffers in a buffer pool

If a buffer pool is too small, the condition can result in message [CSQP020E](#) on the console, you can allocate more buffers to it using the ALTER BUFFPOOL command as follows:

1. Determine how much space is available for new buffers by looking at the [CSQY220I](#) messages in the log. The available space is reported in MB. As a buffer has a size of 4 KB, each MB of available space allows you to allocate 256 buffers. Do not allocate all the free space to buffers, as some is required for other tasks.

If the buffer pool uses fixed 4 KB pages, that is, its PAGECLAS attribute is FIXED4KB, ensure that there is sufficient real storage available on the LPAR.

2. If the reported free space is inadequate, release some buffers from another buffer pool using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool from which you want to reclaim space and *integer* is the new number of buffers to be allocated to this buffer pool, which must be smaller than the original number of buffers allocated to it.

3. Add buffers to the buffer pool you want to expand using the command

```
ALTER BUFFPOOL(buf-pool-id) BUFFERS(integer)
```

where *buf-pool-id* is the buffer pool to be expanded and *integer* is the new number of buffers to be allocated to this buffer pool, which must be larger than the original number of buffers allocated to it.

## How to delete a buffer pool

When a buffer pool is no longer used by any page sets, delete it to release the virtual storage allocated to it.

You delete a buffer pool using the [DELETE BUFFPOOL](#) command. The command fails if any page sets are using this buffer pool.

See [“How to delete page sets” on page 539](#) for information about how to delete page sets.

## Managing queue sharing groups and shared queues on z/OS

IBM MQ can use different types of shared resources, for example queue sharing groups, shared queues, and the coupling facility. Use this topic to review the procedures needed to manage these shared resources.

This section contains information about the following topics:

- [“Managing queue sharing groups” on page 541](#)
- [“Managing shared queues” on page 544](#)
- [“Managing group objects” on page 548](#)
- [“Managing the coupling facility” on page 549](#)

## Managing queue sharing groups

You can add or remove a queue manager to a queue sharing group (QSG), and manage the associated Db2 tables.

This topic has sections about the following tasks:

- [“Setting up a queue sharing group” on page 541](#)
- [“Adding a queue manager to a queue sharing group” on page 542](#)
- [“Removing a queue manager from a queue sharing group” on page 543](#)
- [“Removing a queue sharing group from the Db2 tables” on page 543](#)
- [“Validating the consistency of Db2 definitions” on page 544](#)

### Setting up a queue sharing group

Each queue sharing group has a name of up to four characters. The name must be unique in your network, and must be different from any queue manager names.

Follow these steps to set up a queue sharing group:

1. If this is the first queue sharing group to use the Db2 data-sharing group, [set up the Db2 environment](#).
2. [Set up the coupling facility](#).
3. Add the queue sharing group to the Db2 tables. Use the ADD QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQS).
4. Add a queue manager to the queue sharing group by following the steps in [“Adding a queue manager to a queue sharing group” on page 542](#)
5. Define application structures to IBM MQ by following the steps in [“Adding a coupling facility structure” on page 549](#).
6. If required, [migrate non-shared queues to shared queues](#).
7. For availability, create shared channels into and out of the queue sharing group.
  - For connections into the queue sharing group:
    - Set up a VIPA socket or hardware router to distribute workload between the available queue managers in the QSG.
    - Define a receiver channel with QSGDISP(GROUP), to ensure the channel definition is available on all queue managers in the QSG.
    - Start a listener with INDISP(GROUP), on each queue manager, for MCA channel connections into the QSG. Client connections into the QSG should still connect to a listener started with INDISP(QMGR).
    - Change applications to connect using the QSG name, rather than a specific queue manager name.

- Ensure that the channel authentication rules on all queue managers in the QSG are the same, to allow applications to connect to any queue manager in the QSG.
- For connections out of the queue sharing group:
  - Define a shared transmission queue.
  - Define the outbound channel with QSGDISP(GROUP) and DEFCDISP(SHARED).

If you convert an existing channel to a shared channel, you might need to issue the `RESET CHANNEL` command before starting the channel as the synchronization queue used by the channel will have changed.

## Adding a queue manager to a queue sharing group

A queue manager can be added to an existing queue sharing group.

Note that:

- The queue sharing group must exist before you can add queue managers to it.
- A queue manager can be a member of only one queue sharing group.

Follow these steps to add a queue manager to a queue sharing group:

1. Perform the tasks in [implement ESM security controls for the queue sharing group](#) to grant the appropriate access to the queue manager and channel initiator user IDs.
2. If the queue sharing group has CF structures configured to offload data to SMDS, perform the tasks in [set up the SMDS environment](#).
3. Stop the queue manager.
4. Use the ADD QMGR function of the queue sharing group utility (CSQ5PQSG). This program is described in the [queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45AQM).
5. [Change your system parameter module](#) to add queue sharing group data:
  - a. Modify CSQ6SYSP to specify the QSGDATA parameter. See [using CSQ6SYSP](#) for more information.
  - b. Assemble and link the system parameter module. You might want to use a different name for the load module.
  - c. Change your startup process to use the new module.
6. Copy and tailor sample member thlqual.SCSQPROC(CSQ4INSS), which defines required CF structures and SYSTEM queues. Add the customized member to the CSQINP2 DD in the queue manager startup JCL.
7. Restart your queue manager using the queue sharing group system parameter module.
8. Optionally, migrate to security profiles prefixed by the queue sharing group name, instead of the queue manager name.
9. If shared channels are used for connections into the QSG, create channel authentication rules that mirror those on the other queue managers in the QSG, to allow applications to connect to any queue manager in the QSG.
10. 10. Optionally, do either of the following to allow applications connected to the queue manager in the QSG to put messages to queues hosted by other queue managers in the QSG:
  - Turn on [intra-group queuing](#) by issuing the command `ALTER QMGR IGQ(ENABLED)`.
  - Define transmission queues and channels to the other queue managers in the QSG. Defining transmission queues with the same name as the target queue managers avoids the need to define remote queues and queue manager aliases.

**Note:** To add a queue manager to an existing queue sharing group containing queue managers running earlier versions of IBM MQ, you must first apply the coexistence PTF for the highest version of IBM MQ in the group to every earlier version queue manager in the group.

## Removing a queue manager from a queue sharing group

You can only remove a queue manager from a queue sharing group if the queue manager's logs are not needed by another process, and all SMDS owned by the queue manager are empty.

See [Deleting shared message data sets](#) and [DELETE CFSTRUCT](#) for more information.

The logs are needed if they contain:

- The latest backup of one of the coupling facility (CF) application structures used by the queue sharing group
- Data needed by a future restore process, that is, the queue manager has used a recoverable structure since the time described by the last backup exclusion interval value.

If either or both of these points apply, or an SMDS owned by the queue manager contains messages, the queue manager cannot be removed. To determine which queue managers' logs are needed for a future restore process, use the MQSC DISPLAY CFSTATUS command with the TYPE(BACKUP) option (for details of this command, see [DISPLAY CFSTATUS](#) ).

Use the following steps to remove a queue manager from a queue sharing group:

1. Stop any applications connected to the queue manager that put messages to shared queues.
2. Resolve any indoubt units of work involving this queue manager.
3. Determine if there are any messages in any SMDS owned by the queue manager by issuing the command DISPLAY USAGE TYPE(SMDS).
4. If there are offloaded messages for any application structure, wait until those messages have been retrieved from the queue. The number of offloaded messages reported by DISPLAY USAGE TYPE(SMDS) should be zero before proceeding.
5. Shut the queue manager down cleanly using STOP QMGR MODE(QUIESCE).
6. Wait for an interval at least equivalent to the value of the EXCLINT parameter you will specify in the BACKUP CFSTRUCT command in the next step.
7. On another queue manager, run a CF structure backup for each recoverable CF structure by using the MQSC BACKUP CFSTRUCT command and specifying an EXCLINT value as required in the previous step.
8. Confirm that the queue manager's logs are not needed to restore any CF structures, by inspecting the output from the command DISPLAY CFSTATUS(\*) TYPE(BACKUP).
9. Use the REMOVE QMGR function of the CSQ5PQSG utility to remove the queue manager from the queue sharing group. This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQM).
10. Before restarting the queue manager, reset the QSGDATA system parameter to its default value, and recreate the system parameter module. See [Using CSQ6SYSP](#) for information about how to tailor your system parameters.

Note, that when removing the last queue manager in a queue sharing group, you must use the FORCE option, rather than REMOVE. This removes the queue manager from the queue sharing group, while not performing the consistency checks of the queue manager logs being required for recovery. You should only perform this operation if you are deleting the queue sharing group.

## Removing a queue sharing group from the Db2 tables

To remove a queue sharing group from the Db2 tables, use the REMOVE QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#). A sample is provided in thlqual.SCSQPROC(CSQ45RQS).

You can only remove a queue sharing group from the common Db2 data-sharing group tables after you have removed all the queue managers from the queue sharing group (as described in [“Removing a queue manager from a queue sharing group”](#) on page 543 ).

When the queue sharing group record is deleted from the queue sharing group administration table, all objects and administrative information relating to that queue sharing group are deleted from other IBM MQ Db2 tables. This includes shared queue and group object information.

## Validating the consistency of Db2 definitions

Problems for shared queues within a queue sharing group can occur if the Db2 object definitions have, for any reason, become inconsistent.

To validate the consistency of the Db2 object definitions for queue managers, CF structures, and shared queues, use the VERIFY QSG function of the queue sharing group utility (CSQ5PQSG). This program is described in [The queue sharing group utility](#).

## **Managing shared queues**

Use this topic to understand how to recover, move, and migrate shared queues.

This section describes the following tasks:

- [“Recovering shared queues” on page 544](#)
- [“Moving shared queues” on page 545](#)
- [“Migrating non-shared queues to shared queues” on page 547](#)
- [Suspending a Db2 connection](#)

## Recovering shared queues

IBM MQ can recover persistent messages on shared queues if all:

- Backups of the CF structures containing the messages have been performed.
- All the logs for all queue managers in the queue sharing group are available, to perform recovery from the point the backups are taken.
- Db2 is available and the structure backup table is more recent than the most recent CF structure backup.

The messages on a shared queue are stored in a coupling facility (CF) structure. Persistent messages can be put onto shared queues, and like persistent messages on non-shared queues, they are copied to the queue manager log. The MQSC [BACKUP CFSTRUCT](#) and [RECOVER CFSTRUCT](#) commands are provided to allow the recovery of a CF structure in the unlikely event of a coupling facility failure. In such circumstances, any nonpersistent messages stored in the affected structure are lost, but persistent messages can be recovered. Any further application activity using the structure is prevented until the structure has been recovered.

To enable recovery, you must back up your coupling facility list structures frequently using the MQSC [BACKUP CFSTRUCT](#) command. The messages in the CF structure are written onto the active log data set of the queue manager making the backup. It writes a record of the backup to Db2: the name of the CF structure being backed up, the name of the queue manager doing the backup, the RBA range for this backup on that queue manager log, and the backup time. Back up CF list structures even if you are not actively using shared queues, for example, if you have set up a queue sharing group intending to use it in the future.

You can recover a CF structure by issuing an MQSC [RECOVER CFSTRUCT](#) command to the queue manager that can perform the recovery; you can use any queue manager in the queue sharing group. You can specify a single CF structure to be recovered, or you can recover several CF structures simultaneously.

As noted previously, it is important that you back up your CF list structures frequently, otherwise recovering a CF structure can take a long time. Moreover, the recovery process cannot be canceled.

The definition of a shared queue is kept in a Db2 database and can therefore be recovered if necessary using standard Db2 database procedures. See [Shared queues and queue sharing groups](#) for more information.



## Moving shared queues

This section describes how to perform load balancing by moving a shared queue from one coupling facility structure to another. It also describes how to move a non-shared queue to a shared queue, and how to move a shared queue to a non-shared queue.

When you move a queue, you need to define a temporary queue as part of the procedure. This is because every queue must have a unique name, so you cannot have two queues of the same name, even if the queues have different queue dispositions. IBM MQ tolerates having two queues with the same name (as in step “2” on page 545 ), but you cannot use the queues.

- Moving a queue from one coupling facility structure to another
- Moving a non-shared queue to a shared queue
- Moving a shared queue to a non-shared queue

### Moving a queue from one coupling facility structure to another

To move queues and their messages from one CF structure to another, use the MQSC [MOVE QLOCAL](#) command. When you have identified the queue or queues that you want to move to a new CF structure, use the following procedure to move each queue:

1. Ensure that the queue you want to move is not in use by any applications, that is, the queue attributes IPPROCS and OPPROCS are zero on all queue managers in the queue sharing group.
2. Prevent applications from putting messages on the queue being moved by altering the queue definition to disable MQPUT s. Change the queue definition to PUT(DISABLED).
3. Define a temporary queue with the same attributes as the queue that is being moved using the following command:

```
DEFINE QL(TEMP_QUEUE) LIKE(Queue_To_Move) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
```

**Note:** If this temporary queue exists from a previous run, delete it before doing the define.

4. Move the messages to the temporary queue using the following command:

```
MOVE QLOCAL(Queue_To_Move) TOQLOCAL(TEMP_QUEUE)
```

5. Delete the queue you are moving, using the command:

```
DELETE QLOCAL(Queue_To_Move)
```

6. Redefine the queue that is being moved, changing the CFSTRUCT attribute, using the following command:

```
DEFINE QL(Queue_To_Move) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
```

When the queue is redefined, it is based on the temporary queue created in step “3” on page 545.

7. Move the messages back to the new queue using the command:

```
MOVE QLOCAL(TEMP) TOQLOCAL(Queue_To_Move)
```

8. The queue created in step “3” on page 545 is no longer required. Use the following command to delete it:

```
DELETE QL(TEMP_QUEUE)
```

9. If the queue being moved was defined in the CSQINP2 data sets, change the CFSTRUCT attribute of the appropriate DEFINE QLOCAL command in the CSQINP2 data sets. Add the REPLACE keyword so that the existing queue definition is replaced.

Figure 30 on page 546 shows a sample job for moving a queue from one CF structure to another.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//      DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 30. Sample job for moving a queue from one CF structure to another

### Moving a non-shared queue to a shared queue

The procedure for moving a non-shared queue to a shared queue is like the procedure for moving a queue from one CF structure to another (see “Moving a queue from one coupling facility structure to another” on page 545). Figure 31 on page 546 gives a sample job to do this.

**Note:** Remember that messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=thlqual.SCSQANLE,DISP=SHR
//      DD DSN=thlqual.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 31. Sample job for moving a non-shared queue to a shared queue

## Moving a shared queue to a non-shared queue

The procedure for moving a shared queue to a non-shared queue is like the procedure for moving a queue from one CF structure to another (see [“Moving a queue from one coupling facility structure to another”](#) on page 545).

Figure 32 on page 547 gives a sample job to do this.

```
//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
ALTER QL(Queue_TO_MOVE) PUT(DISABLED)
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED) QSGDISP(QMGR)
MOVE QLOCAL(Queue_TO_MOVE) TOQLOCAL(TEMP_QUEUE)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) STGCLASS(NEW) QSGDISP(QMGR)
MOVE QLOCAL(TEMP_QUEUE) TOQLOCAL(Queue_TO_MOVE)
DELETE QL(TEMP_QUEUE)
/*
```

Figure 32. Sample job for moving a shared queue to a non-shared queue

## Migrating non-shared queues to shared queues

There are two stages to migrating non-shared queues to shared queues:

- Migrating the first (or only) queue manager in the queue sharing group
- Migrating any other queue managers in the queue sharing group

### Migrating the first (or only) queue manager in the queue sharing group

Figure 31 on page 546 shows an example job for moving a non-shared queue to a shared queue. Do this for each queue that needs migrating.

#### Note:

1. Messages on shared queues are subject to certain restrictions on the maximum message size, message persistence, and queue index type, so you might not be able to move some non-shared queues to a shared queue.
2. You must use the correct index type for shared queues. If you migrate a transmission queue to be a shared queue, the index type must be MSGID.

If the queue is empty, or you do not need to keep the messages that are on it, migrating the queue is simpler. Figure 33 on page 548 shows an example job to use in these circumstances.

```

//UTILITY EXEC PGM=CSQUTIL,PARM=('CSQ1')
//STEPLIB DD DSN=th1qua1.SCSQANLE,DISP=SHR
// DD DSN=th1qua1.SCSQAUTH,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
COMMAND DDNAME(MOVEQ)
/*
//MOVEQ DD *
DELETE QL(TEMP_QUEUE) PURGE
DEFINE QL(TEMP_QUEUE) LIKE(Queue_TO_MOVE) PUT(ENABLED) GET(ENABLED)
DELETE QL(Queue_TO_MOVE)
DEFINE QL(Queue_TO_MOVE) LIKE(TEMP_QUEUE) CFSTRUCT(NEW) QSGDISP(SHARED)
DELETE QL(TEMP_QUEUE)
/*

```

Figure 33. Sample job for moving a non-shared queue without messages to a shared queue

### Migrating any other queue managers in the queue sharing group

1. For each queue that does not have the same name as an existing shared queue, move the queue as described in [Figure 31 on page 546](#) or [Figure 33 on page 548](#).
2. For queues that have the same name as an existing shared queue, move the messages to the shared queue using the commands shown in [Figure 34 on page 548](#).

```

MOVE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR) TOQLOCAL(Queue_TO_MOVE)
DELETE QLOCAL(Queue_TO_MOVE) QSGDISP(QMGR)

```

Figure 34. Moving messages from a non-shared queue to an existing shared queue

### Suspending a connection to Db2

If you want to apply maintenance or service to the Db2 tables or package related to shared queues without stopping your queue manager, you must temporarily disconnect queue managers in the data sharing group (DSG) from Db2.

To do this:

1. Use the MQSC command [SUSPEND QMGR FACILITY](#)( Db2 ).
2. Do the binds.
3. Reconnect to Db2 using the MQSC command [RESUME QMGR FACILITY](#)( Db2 )

Note that there are restrictions on the use of these commands.



**Attention:** While the Db2 connection is suspended, the following operations will not be available. Therefore, you need to do this work during a time when your enterprise is at its least busy.

- Access to Shared queue objects for administration (define, delete,alter)
- Starting shared channels
- Storing messages in Db2
- Backup or recover CFSTRUCT

### Managing group objects

Use this topic to understand how to work with group objects.

IBM MQ automatically copies the definition of a group object to page set zero of each queue manager that uses it. You can alter the copy of the definition temporarily, and IBM MQ allows you to refresh the page set copies from the repository copy. IBM MQ always tries to refresh the page set copies from the repository

copy on start-up (for channel objects, this is done when the channel initiator restarts). This ensures that the page set copies reflect the version on the repository, including any changes that were made when the queue manager was inactive.

There are circumstances under which the refresh is not performed, for example:

- If a copy of the queue is open, a refresh that would change the usage of the queue fails.
- If a copy of a queue has messages on it, a refresh that would delete that queue fails.

In these circumstances, the refresh is not performed on that copy, but is performed on the copies on all other queue managers. Check for and correct any problems with copy objects after adding, changing, or deleting a group object, and at queue manager or channel initiator restart.

## **Managing the coupling facility**

Use this topic to understand how to add or remove coupling facility (CF) structures.

This section describes the following tasks:

- [“Adding a coupling facility structure” on page 549](#)
- [“Removing a coupling facility structure” on page 549](#)

### **Adding a coupling facility structure**

To add a coupling facility structure, use the following procedure:

1. Define the CF structure in the CFRM policy data set. The information about setting up the coupling facility in [Set up the coupling facility](#) describes the rules for naming coupling facility structures, and how to define structures in the CFRM policy data set.
2. If you want to configure the structure to offload message data to SMDS, allocate and preformat data sets. See [creating a shared message data set](#) for details.
3. Define the structure to IBM MQ using the [DEFINE CFSTRUCT](#) command.

### **Removing a coupling facility structure**

To remove a coupling facility structure, use the following procedure:

1. Use the following command to get a list of all the queues using the coupling facility structure that you want to delete:

```
DISPLAY QUEUE(*) QSGDISP(SHARED) CFSTRUCT(structure-name)
```

2. Delete all the queues that use the structure.
3. Delete the CF structure from IBM MQ using the [DELETE CFSTRUCT](#) command.
4. If the structure was configured to offload message data to SMDS, delete the SMDS.
5. Remove the structure definition from your CFRM policy data set and run the IXCMIAPU utility. (This is the reverse of the customization task set up the coupling facility, described in [Set up the coupling facility](#).)

## **Tuning coupling facility list monitoring**

Use this topic to understand coupling facility list monitoring

Coupling facility (CF) list monitoring is used to monitor the state of list structures containing IBM MQ shared queues. When a message is added to a shared queue, and the queue's depth transitions from zero to non-zero, the CF notifies all queue managers in the queue sharing group. When notified the

queue managers might perform a number of actions, including notifying trigger monitors that are using TRIGGER(FIRST), or applications which are performing a get-wait.

By default, the CF notifies all queue managers in the queue sharing group at the same time. In certain configurations this can cause problems, such as:

- Skewed workload distribution, where a large percentage of messages go to a particular queue manager in the queue sharing group, often the queue manager running on the fastest LPAR, or which is closest to the CF, or
- A large number of failed gets, resulting in wasted CPU time.

z/OS V2R3 introduces a new coupling facility resource manager (CFRM) attribute called **KEYRNOTIFYDELAY**, which can be used with list structures containing shared queues (that is, application structures, and not the admin structure), and which can, for certain workloads, minimize the effects of workload skewing and empty MQGET calls, or empty MQGET calls.

**KEYRNOTIFYDELAY** can only be set on structures in a CF, running at CFLEVEL 22 or higher.

Its value must be one to seven decimal digits, in a range from 0 to 1,000,000 microseconds. If set to a non-zero value and the depth of a queue transitions from zero to non-zero, the CF selects a single queue manager from the queue sharing group, and notifies that queue manager before all the other queue managers in the group.

The queue manager is selected in a round-robin manner. If the selected queue manager does not process the message inside the time interval described by **KEYRNOTIFYDELAY** all the other queue managers in the queue sharing group will also be notified.

More information on **KEYRNOTIFYDELAY** is available here: [Understanding Keyrange Monitoring Notification Delay](#).

Note that there are two similar CFRM attributes called **LISTNOTIFYDELAY** and **SUBNOTIFYDELAY**. Neither of these has any measurable effect on IBM MQ workload.

## Recovery and restart on z/OS

Use this topic to understand the recovery and restart mechanisms used by IBM MQ.

### Restarting IBM MQ

After a queue manager terminates there are different restart procedures needed depending on how the queue manager terminated. Use this topic to understand the different restart procedures that you can use.

This topic contains information about how to restart your queue manager in the following circumstances:

- [“Restarting after a normal shutdown” on page 550](#)
- [“Restarting after an abnormal termination” on page 551](#)
- [“Restarting if you have lost your page sets” on page 551](#)
- [“Restarting if you have lost your log data sets” on page 551](#)
- [Restarting if you have lost your CF structures](#)

### Restarting after a normal shutdown

If the queue manager was stopped with the STOP QMGR command, the system finishes its work in an orderly way and takes a termination checkpoint before stopping. When you restart the queue manager, it uses information from the system checkpoint and recovery log to determine the system status at shutdown.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 476](#).

## Restarting after an abnormal termination

IBM MQ automatically detects whether restart follows a normal shutdown or an abnormal termination.

Starting the queue manager after it has terminated abnormally is different from starting it after the STOP QMGR command has been issued. If the queue manager terminates abnormally, it terminates without being able to finish its work or take a termination checkpoint.

To restart the queue manager, issue the START QMGR command as described in [“Using MQSC to start and stop a queue manager on z/OS” on page 476](#). When you restart a queue manager after an abnormal termination, it refreshes its knowledge of its status at termination using information in the log, and notifies you of the status of various tasks.

Normally, the restart process resolves all inconsistent states. But, in some cases, you must take specific steps to resolve inconsistencies. This is described in [“Recovering units of work manually” on page 563](#).

## Restarting if you have lost your page sets

If you have lost your page sets, you need to restore them from your backup copies before you can restart the queue manager. This is described in [“How to back up and recover page sets” on page 536](#).

The queue manager might take a long time to restart under these circumstances because of the length of time needed for media recovery.

## Restarting if you have lost your log data sets

If, after stopping a queue manager (using the STOP QMGR command), both copies of the log are lost or damaged, you can restart the queue manager providing you have a consistent set of page sets (produced using [Method 1: Full backup](#)).

Follow this procedure:

1. Define new page sets to correspond to each existing page set in your queue manager. See [Task 15: Define your page sets](#) for information about page set definition.  
Ensure that each new page set is larger than the corresponding source page set.
2. Use the FORMAT function of CSQUTIL to format the destination page set. See [Formatting page sets](#) for more details.
3. Use the RESETPAGE function of CSQUTIL to copy the existing page sets or reset them in place, and reset the log RBA in each page. See [Copying a page set and resetting the log](#) for more information about this function.
4. Redefine your queue manager log data sets and BSDS using CSQJU003 (see [The change log inventory utility](#)).
5. Restart the queue manager using the new page sets. To do this, you do one of the following:
  - Change the queue manager started task procedure to reference the new page sets. See [Task 6: Create procedures for the IBM MQ queue manager](#) for more information.
  - Use Access Method Services to delete the old page sets and then rename the new page sets, giving them the same names as the old page sets.

**Attention:** Before you delete any IBM MQ page set, ensure that you have made the required backup copies.

If the queue manager is a member of a queue sharing group, GROUP and SHARED object definitions are not normally affected by lost or damaged logs. However, if any shared-queue messages are involved in a unit of work that was covered by the lost or damaged logs, the effect on such uncommitted messages is unpredictable.

**Note:** If logs are damaged and the queue manager is a member of a queue sharing group, the ability to recover shared persistent messages might be lost. Issue a BACKUP CFSTRUCT command immediately on another active queue manager in the queue sharing group for all CF structures with the RECOVER(YES) attribute.

## Restarting if you have lost your CF structures

You do not need to restart if you lose your CF structures, because the queue manager does not terminate.

### **Alternative site recovery on z/OS**

You can recover a single queue manager or a queue sharing group, or consider disk mirroring.

See the following sections for more details:

- [Recovering a single queue manager at an alternative site](#)
- [Recovering a queue sharing group.](#)
  - [CF structure media recovery](#)
  - [Backing up the queue sharing group at the prime site](#)
  - [Recovering a queue sharing group at the alternative site](#)
- [Using disk mirroring](#)

## Recovering a single queue manager at an alternative site

If a total loss of an IBM MQ computing center occurs, you can recover on another queue manager or queue sharing group at a recovery site. (See “[Recovering a queue sharing group at the alternative site](#)” on page 555 for the alternative site recovery procedure for a queue sharing group.)

To recover on another queue manager at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time, as possible.

At the recovery site:

- The recovery queue managers **must** have the same names as the lost queue managers.
- The system parameter module (for example, CSQZPARM) used on each recovery queue manager must contain the same parameters as the corresponding lost queue manager.

When you have done this, reestablish all your queue managers as described in the following procedure. This can be used to perform disaster recovery at the recovery site for a single queue manager. It assumes that all that is available are:

- Copies of the archive logs and BSDSs created by normal running at the primary site (the active logs will have been lost along with the queue manager at the primary site).
- Copies of the page sets from the queue manager at the primary site that are the same age or older than the most recent archive log copies available.

You can use dual logging for the active and archive logs, in which case you need to apply the BSDS updates to both copies:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA and ENDRBA of this log.



5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the STARTRBA and ENDRBA recorded in Step “4” on page 552.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Use CSQJU003 to add a restart control record to the BSDS. Specify CRESTART CREATE, ENDRBA=highrba, where highrba is the high RBA of the most recent archive log available (found in Step “4” on page 552 ), plus 1.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

9. Restart the queue manager with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.
REPLY Y TO CONTINUE, N TO CANCEL
```

Type Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

See [IBM MQ utilities on z/OS reference](#) for information about using CSQJU003 and CSQJU004.

The following example shows sample input statements for CSQJU003 for steps 6, 7, and 8:

```
* Step 6
DELETE DSNAME=MQM2.LOGCOPY1.DS01
DELETE DSNAME=MQM2.LOGCOPY1.DS02
DELETE DSNAME=MQM2.LOGCOPY1.DS03
DELETE DSNAME=MQM2.LOGCOPY1.DS04
DELETE DSNAME=MQM2.LOGCOPY2.DS01
DELETE DSNAME=MQM2.LOGCOPY2.DS02
DELETE DSNAME=MQM2.LOGCOPY2.DS03
DELETE DSNAME=MQM2.LOGCOPY2.DS04

* Step 7
NEWLOG DSNAME=MQM2.LOGCOPY1.DS01,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS02,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS03,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY1.DS04,COPY1
NEWLOG DSNAME=MQM2.LOGCOPY2.DS01,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS02,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS03,COPY2
NEWLOG DSNAME=MQM2.LOGCOPY2.DS04,COPY2

* Step 8
CRESTART CREATE, ENDRBA=063000
```

The things you need to consider for restarting the channel initiator at the recovery site are like those faced when using ARM to restart the channel initiator on a different z/OS image. See [“Using ARM in an IBM MQ network” on page 561](#) for more information. Your recovery strategy should also cover recovery of the IBM MQ product libraries and the application programming environments that use IBM MQ ( CICS , for example).

Other functions of the change log inventory utility (CSQJU003) can also be used in disaster recovery scenarios. The HIGHRBA function allows the update of the highest RBA written and highest RBA offloaded values within the bootstrap data set. The CHECKPT function allows the addition of new checkpoint queue records or the deletion of existing checkpoint queue records in the BSDS.

**Attention: These functions might affect the integrity of your IBM MQ data.** Only use them in disaster recovery scenarios under the guidance of IBM service personnel.

### Fast copy techniques

If copies of all the page sets and logs are made while the queue manager is frozen, the copies will be a consistent set that can be used to restart the queue manager at an alternative site. They typically enable a much faster restart of the queue manager, as there is little media recovery to be performed.

Use the SUSPEND QMGR LOG command to freeze the queue manager. This command flushes buffer pools to the page sets, takes a checkpoint, and stops any further log write activity. Once log write activity has been suspended, the queue manager is effectively frozen until you issue a RESUME QMGR LOG command. While the queue manager is frozen, the page sets and logs can be copied.

By using copying tools such as FLASHCOPY or SNAPSHOT to rapidly copy the page sets and logs, the time during which the queue manager is frozen can be reduced to a minimum.

Within a queue sharing group, however, the SUSPEND QMGR LOG command might not be such a good solution. To be effective, the copies of the logs must all contain the same point in time for recovery, which means that the SUSPEND QMGR LOG command must be issued on all queue managers within the queue sharing group simultaneously, and therefore the entire queue sharing group will be frozen for some time.

## Recovering a queue sharing group

In the event of a prime site disaster, you can restart a queue sharing group at a remote site using backup data sets from the prime site. To recover a queue sharing group you need to coordinate the recovery across all the queue managers in the queue sharing group, and coordinate with other resources, primarily Db2. This section describes these tasks in detail.

- [CF structure media recovery](#)
- [Backing up the queue sharing group at the prime site](#)
- [Recovering a queue sharing group at the alternative site](#)

### CF structure media recovery

Media recovery of a CF structure used to hold persistent messages on a shared queue, relies on having a backup of the media that can be forward recovered by the application of logged updates. Take backups of your CF structures periodically using the MQSC BACKUP CFSTRUCT command. All updates to shared queues (MQGETs and MQPUTs) are written on the log of the queue manager where the update is performed. To perform media recovery of a CF structure you must apply logged updates to that backup from the logs of all the queue managers that have used that CF structure. When you use the MQSC RECOVER CFSTRUCT command, IBM MQ automatically merges the logs from relevant queue managers, and applies the updates to the most recent backup.

The CF structure backup is written to the log of the queue manager that processed the BACKUP CFSTRUCT command, so there are no additional data sets to be collected and transported to the alternative site.

### Backing up the queue sharing group at the prime site

At the prime site you need to establish a consistent set of backups on a regular basis, which can be used in the event of a disaster to rebuild the queue sharing group at an alternative site. For a single queue manager, recovery can be to an arbitrary point in time, typically the end of the logs available at the remote site. However, where persistent messages have been stored on a shared queue, the logs of all the queue managers in the queue sharing group must be merged to recover shared queues, as any queue manager in the queue sharing group might have performed updates (MQPUTs or MQGETs) on the queue.

For recovery of a queue sharing group, you need to establish a point in time that is within the log range of the log data of all queue managers. However, as you can only **forward** recover media from the log, this point in time must be after the BACKUP CFSTRUCT command has been issued and after any page set backups have been performed. (Typically, the point in time for recovery might correspond to the end of a business day or week.)

The following diagram shows time lines for two queue managers in a queue sharing group. For each queue manager, fuzzy backups of page sets are taken (see [Method 2: Fuzzy backup](#)). On queue manager A, a BACKUP CFSTRUCT command is issued. Subsequently, an ARCHIVE LOG command is issued on each queue manager to truncate the active log, and copy it to media offline from the queue manager, which can be transported to the alternative site. End of log identifies the time at which the ARCHIVE LOG command was issued, and therefore marks the extent of log data typically available at the alternative site. The point in time for recovery must lie between the end of any page set or CF structure backups, and the earliest end of log available at the alternative site.

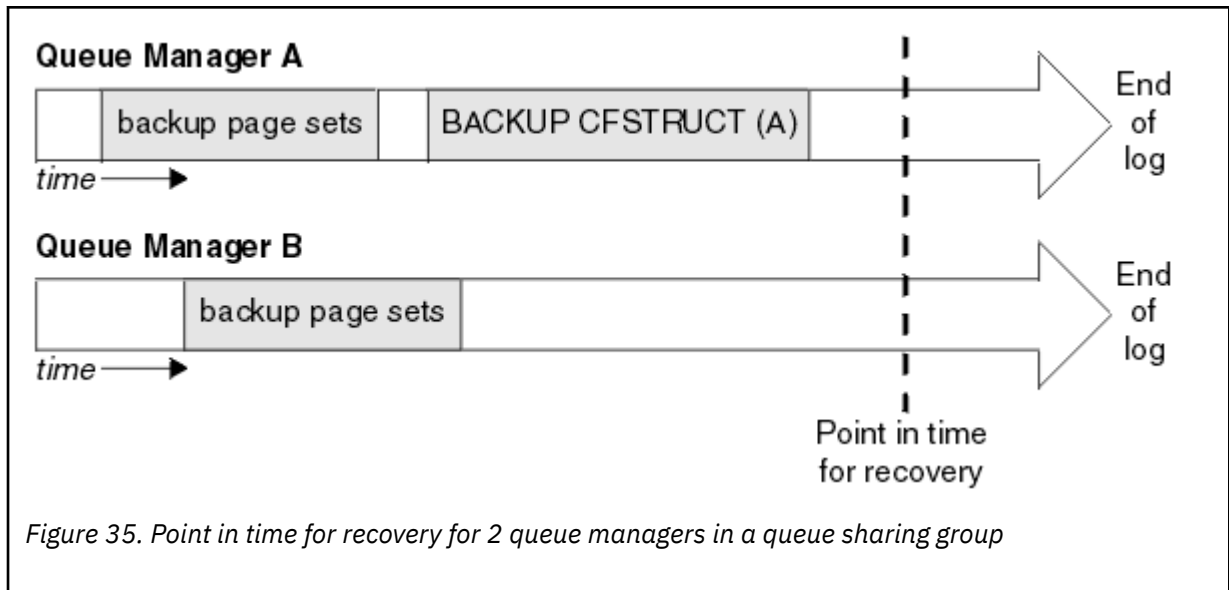


Figure 35. Point in time for recovery for 2 queue managers in a queue sharing group

IBM MQ records information associated with the CF structure backups in a table in Db2. Depending on your requirements, you might want to coordinate the point in time for recovery of IBM MQ with that for Db2, or it might be sufficient to take a copy of the IBM MQ CSQ.ADMIN\_B\_STRBACKUP table after the BACKUP CFSTRUCT commands have finished.

To prepare for a recovery:

1. Create page set backups for each queue manager in the queue sharing group.
2. Issue a BACKUP CFSTRUCT command for each CF structure with the RECOVER(YES) attribute. You can issue these commands from a single queue manager, or from different queue managers within the queue sharing group to balance the workload.
3. Once all the backups have completed, issue an ARCHIVE LOG command to switch the active log and create copies of the logs and BSDS of each queue manager in the queue sharing group.
4. Transport the page set backups, the archived logs, the archived BSDS of all the queue managers in the queue sharing group, and your chosen Db2 backup information, off-site.

### Recovering a queue sharing group at the alternative site

Before you can recover the queue sharing group, you need to prepare the environment:

1. If you have old information in your coupling facility from practice startups when you installed the queue sharing group, you need to clean this out first:

**Note:** If you do not have old information in the coupling facility, you can omit this step.

- a. Enter the following z/OS command to display the CF structures for this queue sharing group:

```
D XCF,STRUCTURE,STRNAME= qsgname
```

- b. For all structures that start with the queue sharing group name, use the z/OS command SETXCF FORCE CONNECTION to force the connection off those structures:

```
SETXCF FORCE,CONNECTION,STRNAME= strname,CONNAME=ALL
```

- c. Delete all the CF structures using the following command for each structure:

```
SETXCF FORCE,STRUCTURE,STRNAME= strname
```

2. Restore Db2 systems and data-sharing groups.
3. Recover the CSQ.ADMIN\_B\_STRBACKUP table so that it contains information about the most recent structure backups taken at the prime site.

**Note:** It is important that the STRBACKUP table contains the most recent structure backup information. Older structure backup information might require data sets that you have discarded as a result of the information given by a recent DISPLAY USAGE TYPE(DATASET) command, which would mean that your recovered CF structure would not contain accurate information.

4. Run the ADD QMGR command of the CSQ5PQSG utility for every queue manager in the queue sharing group. This will restore the XCF group entry for each queue manager.

When you run the utility in this scenario, the following messages are normal:

```
CSQU566I Unable to get attributes for admin structure, CF not found  
or not allocated  
CSQU546E Unable to add QMGR queue_manager_name entry,  
already exists in DB2 table CSQ.ADMIN_B_QMGR  
CSQU148I CSQ5PQSG Utility completed, return code=4
```

To recover the queue managers in the queue sharing group:

1. Define new page set data sets and load them with the data in the copies of the page sets from the primary site.
2. Define new active log data sets.
3. Define a new BSDS data set and use Access Method Services REPRO to copy the most recent archived BSDS into it.
4. Use the print log map utility CSQJU004 to print information from this most recent BSDS. At the time this BSDS was archived, the most recent archived log you have would have just been truncated as an active log, and does not appear as an archived log. Record the STARTRBA, STARTLRSN, ENDRBA, and ENDLRSN values of this log.
5. Use the change log inventory utility, CSQJU003, to register this latest archive log data set in the BSDS that you have just restored, using the values recorded in Step “4” on page 556.
6. Use the DELETE option of CSQJU003 to remove all active log information from the BSDS.
7. Use the NEWLOG option of CSQJU003 to add active logs to the BSDS, do not specify STARTRBA or ENDRBA.
8. Calculate the *recoverylrsn* for the queue sharing group. The *recoverylrsn* is the lowest of the ENDLRSNs across all queue managers in the queue sharing group (as recorded in Step “4” on page 556), minus 1. For example, if there are two queue managers in the queue sharing group, and the ENDLRSN for one of them is B713 3C72 22C5, and for the other is B713 3D45 2123, the *recoverylrsn* is B713 3C72 22C4.
9. Use CSQJU003 to add a restart control record to the BSDS. Specify:

```
CRESTART CREATE,ENDLRSN= recoverylrsn
```

where *recoverylrsn* is the value you recorded in Step “8” on page 556.

The BSDS now describes all active logs as being empty, all the archived logs you have available, and no checkpoints beyond the end of your logs.

You must add the CRESTART record to the BSDS for each queue manager within the queue sharing group.

10. Restart each queue manager in the queue sharing group with the START QMGR command. During initialization, an operator reply message such as the following is issued:

```
CSQJ245D +CSQ1 RESTART CONTROL INDICATES TRUNCATION AT RBA highrba.  
REPLY Y TO CONTINUE, N TO CANCEL
```

Reply Y to start the queue manager. The queue manager starts, and recovers data up to ENDRBA specified in the CRESTART statement.

The first IBM MQ queue manager started can rebuild the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group at this stage.

11. When the admin structure data for all queue managers has been rebuilt, issue a RECOVER CFSTRUCT command for each CF application structure.

If you issue the RECOVER CFSTRUCT command for all structures on a single queue manager, the log merge process is only performed once, so is quicker than issuing the command on a different queue manager for each CF structure, where each queue manager has to perform the log merge step.

When conditional restart processing is used in a queue sharing group, IBM MQ queue managers, performing peer admin rebuild, check that peers BSDS contain the same CRESTART LRSN as their own. This is to ensure the integrity of the rebuilt admin structure. It is therefore important to restart other peers in the QSG, so they can process their own CRESTART information, before the next unconditional restart of any member of the group.

## Using disk mirroring

Many installations now use disk mirroring technologies such as IBM Metro Mirror (formerly PPRC) to make synchronous copies of data sets at an alternative site. In such situations, many of the steps detailed become unnecessary as the IBM MQ page sets and logs at the alternative site are effectively identical to those at the prime site. Where such technologies are used, the steps to restart a queue sharing group at an alternative site may be summarized as:

- Clear IBM MQ CF structures at the alternative site. (These often contain residual information from any previous disaster recovery exercise).
- Restore Db2 systems and all tables in the database used by the IBM MQ queue sharing group.
- Restart queue managers. Before IBM WebSphere MQ 7.0.1, it is necessary to restart each queue manager defined in the queue sharing group as each queue manager recovers its own partition of the admin structure during queue manager restart. After each queue manager has been restarted, those not on their home LPAR can be shut down again. The first IBM MQ queue manager started rebuilds the admin structure partitions for other members of the queue sharing group as well as its own, and it is no longer necessary to restart each queue manager in the queue sharing group.
- After the admin structure has been rebuilt, recover the application structures.

IBM MQ for z/OS supports use of zHyperWrite when writing to active logs mirrored using Metro Mirror. zHyperWrite can help reduce the performance impact of using Metro Mirror; see [Using Metro Mirror with IBM MQ](#) for more information.

### **Reinitializing a queue manager**

If the queue manager has terminated abnormally you might not be able to restart it. This could be because your page sets or logs have been lost, truncated, or corrupted. If this has happened, you might have to reinitialize the queue manager (perform a cold start).

## Attention

**Only perform a cold start if you cannot restart the queue manager any other way.** Performing a cold start enables you to recover your queue manager and your object definitions; you will **not** be able to recover your message data. Check that none of the other restart scenarios described in this topic work for you before you do this.

When you have restarted, all your IBM MQ objects are defined and available for use, but there is no message data.

**Note:** Do not reinitialize a queue manager while it is part of a cluster. You must first remove the queue manager from the cluster (using RESET CLUSTER commands on the other queue managers in the cluster), then reinitialize it, and finally reintroduce it to the cluster as a new queue manager.

This is because during reinitialization, the queue manager identifier (QMID) is changed, so any cluster object with the old queue manager identifier must be removed from the cluster.

For further information see the following sections:

- [Reinitializing a queue manager that is not in a queue sharing group](#)
- [Reinitializing queue managers in a queue sharing group](#)

## Reinitializing a queue manager that is not in a queue sharing group

To reinitialize a queue manager, follow this procedure:

1. Prepare the object definition statements that to be used when you restart the queue manager. To do this, either:
  - If page set zero is available, use the CSQUTIL SDEFS function (see [Producing a list of IBM MQ define commands](#) ). You must get definitions for all object types (authentication information objects, CF structures, channels, namelists, processes, queues, and storage classes).
  - If page set zero is not available, use the definitions from the last time you backed up your object definitions.
2. Redefine your queue manager data sets (do not do this until you have completed step “1” on page [558](#) ).  
See [creating the bootstrap and log data sets](#) and [defining your page sets](#) for more information.
3. Restart the queue manager using the newly defined and initialized log data sets, BSDS, and page sets. Use the object definition input statements that you created in step “1” on page [558](#) as input in the CSQINP2 initialization input data set.

## Reinitializing queue managers in a queue sharing group

In a queue sharing group, reinitializing a queue manager is more complex. It might be necessary to reinitialize one or more queue managers because of page set or log problems, but there might also be problems with Db2 or the coupling facility to deal with. Because of this, there are a number of alternatives:

### Cold start

Reinitializing the entire queue sharing group involves forcing all the coupling facilities structures, clearing all object definitions for the queue sharing group from Db2, deleting or redefining the logs and BSDS, and formatting page sets for all the queue managers in the queue sharing group.

### Shared definitions retained

Delete or redefine the logs and BSDS, format page sets for all queue managers in the queue sharing group, and force all the coupling facilities structures. On restart, all messages will have been deleted. The queue managers re-create COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist and can be used.

### Single queue manager reinitialized

Delete or redefine the logs and BSDS, and format page sets for the single queue manager (this deletes all its private objects and messages). On restart, the queue manager re-creates COPY objects that correspond to GROUP objects that still exist in the Db2 database. Any shared queues still exist, as do the messages on them, and can be used.

### Point in time recovery of a queue sharing group

This is the alternative site disaster recovery scenario.

Shared objects are recovered to the point in time achieved by Db2 recovery (described in [A Db2 system fails](#)). Each queue manager can be recovered to a point in time achievable from the backup copies available at the alternative site.

Persistent messages can be used in queue sharing groups, and can be recovered using the MQSC RECOVER CFSTRUCT command. Note that this command recovers to the time of failure. However, there is no recovery of nonpersistent shared queue messages; they are lost unless you have made backup copies independently using the COPY function of the CSQUTIL utility program.

It is not necessary to try to restore each queue manager to the same point in time because there are no interdependencies between the local objects on different queue managers (which are what is actually being recovered), and the queue manager resynchronization with Db2 on restart creates or deletes COPY objects as necessary on a queue manager by queue manager basis.

## Using the z/OS Automatic Restart Manager (ARM)

Use this topic to understand how you can use ARM to automatically restart your queue managers.

This section contains information about the following topics:

- [“What is the ARM?” on page 559](#)
- [“ARM policies” on page 560](#)
- [“Using ARM in an IBM MQ network” on page 561](#)

### What is the ARM?

The z/OS Automatic Restart Manager (ARM) is a z/OS recovery function that can improve the availability of your queue managers. When a job or task fails, or the system on which it is running fails, ARM can restart the job or task without operator intervention.

If a queue manager or a channel initiator has failed, ARM restarts it on the same z/OS image. If z/OS, and hence a whole group of related subsystems and applications have failed, ARM can restart all the failed systems automatically, in a predefined order, on another z/OS image within the sysplex. This is called a *cross-system restart*.

Restart the channel initiator by ARM only in exceptional circumstances. If the queue manager is restarted by ARM, restart the channel initiator from the CSQINP2 initialization data set (see [“Using ARM in an IBM MQ network” on page 561](#)).

You can use ARM to restart a queue manager on a different z/OS image within the sysplex in the event of z/OS failure. The network implications of IBM MQ ARM restart on a different z/OS image are described in [“Using ARM in an IBM MQ network” on page 561](#).

To enable automatic restart:

- Set up an ARM couple data set.
- Define the automatic restart actions that you want z/OS to perform in an *ARM policy*.
- Start the ARM policy.

Also, IBM MQ must register with ARM at startup (this happens automatically).

**Note:** If you want to restart queue managers in different z/OS images automatically, you must define every queue manager as a subsystem in each z/OS image on which that queue manager might be restarted, with a sysplex wide unique four character subsystem name.

## ARM couple data sets

Ensure that you define the couple data sets required for ARM, and that they are online and active before you start any queue manager for which you want ARM support. IBM MQ automatic ARM registration fails if the couple data sets are not available at queue manager startup. In this situation, IBM MQ assumes that the absence of the couple data set means that you do not want ARM support, and initialization continues.

See *z/OS MVS Setting up a Sysplex* for information about ARM couple data sets.

## ARM policies

The Automatic Restart Manager policies are user-defined rules that control ARM functions that can control any restarts of a queue manager.

ARM functions are controlled by a user-defined *ARM policy*. Each z/OS image running a queue manager instance that is to be restarted by ARM must be connected to an ARM couple data set with an active ARM policy.

IBM provides a default ARM policy. You can define new policies, or override the policy defaults by using the *administrative data utility* (IXCMIAPU) provided with z/OS. *z/OS MVS Setting up a Sysplex* describes this utility, and includes full details of how to define an ARM policy.

Figure 36 on page 560 shows an example of an ARM policy. This sample policy restarts any queue manager within a sysplex, if either the queue manager failed, or a whole system failed.

```
//IXCMIAPU EXEC PGM=IXCMIAPU,REGION=2M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DATA TYPE(ARM)
DEFINE POLICY NAME(ARMPOL1) REPLACE(YES)
RESTART_GROUP(DEFAULT)
ELEMENT(*)
RESTART_ATTEMPTS(0) /* Jobs not to be restarted by ARM */
RESTART_GROUP(GROUP1)
ELEMENT(SYSMQGRMQ*) /* These jobs to be restarted by ARM */
/*
```

Figure 36. Sample ARM policy

For more information see:

- [Defining an ARM policy](#)
- [Activating an ARM policy](#)
- [Registering with ARM](#)

## Defining an ARM policy

Set up your ARM policy as follows:

- Define RESTART\_GROUPS for each queue manager instance that also contain any CICS or IMS subsystems that connect to that queue manager instance. If you use a subsystem naming convention, you might be able to use the '?' and '\*' wild-card characters in your element names to define RESTART\_GROUPS with minimum definition effort.
- Specify TERMTYPE(ELEMTERM) for your channel initiators to indicate that they will be restarted only if the channel initiator has failed and the z/OS image has not failed.



- Specify `TERMTYPE(ALLTERM)` for your queue managers to indicate that they will be restarted if either the queue manager has failed or the z/OS image has failed.
- Specify `RESTART_METHOD(BOTH, PERSIST)` for both queue managers and channel initiators. This tells ARM to restart using the JCL it saved (after resolution of system symbols) during the last startup. It tells ARM to do this irrespective of whether the individual element failed, or the z/OS image failed.
- Accept the default values for all the other ARM policy options.

## Activating an ARM policy

To start your automatic restart management policy, issue the following z/OS command:

```
SETXCF START,POLICY,TYPE=ARM,POLNAME= mypol
```

When the policy is started, all systems connected to the ARM couple data set use the same active policy. Use the `SETXCF STOP` command to disable automatic restarts.

## Registering with ARM

IBM MQ registers automatically as an *ARM element* during queue manager startup (subject to ARM availability). It deregisters during its shutdown phase, unless requested not to.

At startup, the queue manager determines whether ARM is available. If it is, IBM MQ registers using the name `SYSMQMGR ssid`, where *ssid* is the four character queue manager name, and `SYSMQMGR` is the element type.

The `STOP QMGR MODE(QUIESCE)` and `STOP QMGR MODE(FORCE)` commands deregister the queue manager from ARM (if it was registered with ARM at startup). This prevents ARM restarting this queue manager. The `STOP QMGR MODE(RESTART)` command does not deregister the queue manager from ARM, so it is eligible for immediate automatic restart.

Each channel initiator address space determines whether ARM is available, and if so registers with the element name `SYSMQCH ssid`, where *ssid* is the queue manager name, and `SYSMQCH` is the element type.

The channel initiator is always deregistered from ARM when it stops normally, and remains registered only if it ends abnormally. The channel initiator is always deregistered if the queue manager fails.

### Using ARM in an IBM MQ network

You can set up your queue manager so that the channel initiators and associated listeners are started automatically when the queue manager is restarted.

To ensure fully automatic queue manager restart on the same z/OS image for both LU 6.2 and TCP/IP communication protocols:

- Start your listeners automatically by adding the appropriate `START LISTENER` command to the `CSQINPX` data set.
- Start your channel initiator automatically by adding the appropriate `START CHINIT` command to the `CSQINP2` data set.

For restarting a queue manager with TCP/IP or LU6.2, see

- [“Restarting on a different z/OS image with TCP/IP” on page 562](#)
- [“Restarting on a different z/OS image with LU 6.2” on page 563](#)

See [Task 13: Customize the initialization input data sets](#) for information about the `CSQINP2` and `CSQINPX` data sets.

## Restarting on a different z/OS image with TCP/IP

If you are using TCP/IP as your communication protocol, and you are using virtual IP addresses, you can configure these to recover on other z/OS images, allowing channels connecting to that queue manager to reconnect without any changes. Otherwise, you can reallocate a TCP/IP address after moving a queue manager to a different z/OS image only if you are using clusters or if you are connecting to a queue sharing group using a WLM dynamic Domain Name System (DNS) logical group name.

- [When using clustering](#)
- [When connecting to a queue sharing group](#)

### When using clustering

z/OS ARM responds to a system failure by restarting the queue manager on a different z/OS image in the same sysplex; this system has a different TCP/IP address to the original z/OS image. The following explains how you can use IBM MQ clusters to reassign a queue manager's TCP/IP address after it has been moved by ARM restart to a different z/OS image.

When a client queue manager detects the queue manager failure (as a channel failure), it responds by reallocating suitable messages on its cluster transmission queue to a different server queue manager that hosts a different instance of the target cluster queue. However, it cannot reallocate messages that are bound to the original server by affinity constraints, or messages that are in doubt because the server queue manager failed during end-of-batch processing. To process these messages, do the following:

1. Allocate a different cluster-receiver channel name and a different TCP/IP port to each z/OS queue manager. Each queue manager needs a different port so that two systems can share a single TCP/IP stack on a z/OS image. One of these is the queue manager originally running on that z/OS image, and the other is the queue manager that ARM will restart on that z/OS image following a system failure. Configure each port on each z/OS image, so that ARM can restart any queue manager on any z/OS image.
2. Create a different channel initiator command input file (CSQINPX) for each queue manager and z/OS image combination, to be referenced during channel initiator startup.

Each CSQINPX file must include a START LISTENER PORT(port) command specific to that queue manager, and an ALTER CHANNEL command for a cluster-receiver channel specific to that queue manager and z/OS image combination. The ALTER CHANNEL command needs to set the connection name to the TCP/IP name of the z/OS image on which it is restarted. It must include the port number specific to the restarted queue manager as part of the connection name.

The start-up JCL of each queue manager can have a fixed data set name for this CSQINPX file, and each z/OS image must have a different version of each CSQINPX file on a non-shared DASD volume.

If an ARM restart occurs, IBM MQ advertises the changed channel definition to the cluster repository, which in turn publishes it to all the client queue managers that have expressed an interest in the server queue manager.

The client queue manager treats the server queue manager failure as a channel failure, and tries to restart the failed channel. When the client queue manager learns the new server connection-name, the channel restart reconnects the client queue manager to the restarted server queue manager. The client queue manager can then resynchronize its messages, resolve any in-doubt messages on the client queue manager's transmission queue, and normal processing can continue.

### When connecting to a queue sharing group

When connecting to a queue sharing group through a TCP/IP dynamic Domain Name System (DNS) logical group name, the connection name in your channel definition specifies the logical group name of your queue sharing group, not the host name or IP address of a physical machine. When this channel starts, it connects to the dynamic DNS and is then connected to one of the queue managers in the queue sharing group. This process is explained in [Setting up communication for IBM MQ for z/OS using queue sharing groups](#).

In the unlikely event of an image failure, one of the following occurs:

- The queue managers on the failing image de-register from the dynamic DNS running on your sysplex. The channel responds to the connection failure by entering RETRYING state and then connects to the dynamic DNS running on the sysplex. The dynamic DNS allocates the inbound request to one of the remaining members of the queue sharing group that is still running on the remaining images.
- If no other queue manager in the queue sharing group is active and ARM restarts the queue manager and channel initiator on a different image, the group listener registers with dynamic DNS from this new image. This means that the logical group name (from the connection name field of the channel) connects to the dynamic DNS and is then connected to the same queue manager, now running on a different image. No change was required to the channel definition.

For this type of recovery to occur, the following points must be noted:

- On z/OS, the dynamic DNS runs on one of the z/OS images in the sysplex. If this image were to fail, the dynamic DNS needs to be configured so that there is a secondary name server active in the sysplex, acting as an alternative to the primary name server. Information about primary and secondary dynamic DNS servers can be found in the [OS/390® SecureWay CS IP Configuration](#) manual
- The TCP/IP group listener might have been started on a particular IP address that might not be available on this z/OS image. If so, the listener might need to be started on a different IP address on the new image. If you are using virtual IP addresses, you can configure these to recover on other z/OS images so that no change to the START LISTENER command is required.

## Restarting on a different z/OS image with LU 6.2

If you use only LU 6.2 communication protocols, carry out the following procedure to enable network reconnect after automatic restart of a queue manager on a different z/OS image within the sysplex:

- Define each queue manager within the sysplex with a unique subsystem name.
- Define each channel initiator within the sysplex with a unique LUNAME. This is specified in both the queue manager attributes and in the START LISTENER command.

**Note:** The LUNAME names an entry in the APPC side table, which in turn maps this to the actual LUNAME.

- Set up a shared APPC side table, which is referenced by each z/OS image within the sysplex. This should contain an entry for each channel initiator's LUNAME. See [z/OS MVS Planning: APPC/MVS Management](#) for information about this.
- Set up an APPCPM xx member of SYS1.PARMLIB for each channel initiator within the sysplex to contain an LUADD to activate the APPC side table entry for that channel initiator. These members should be shared by each z/OS image. The appropriate SYS1.PARMLIB member is activated by a z/OS command SET APPC= xx, which is issued automatically during ARM restart of the queue manager (and its channel initiator) on a different z/OS image, as described in the following text.
- Use the LU62ARM queue manager attribute to specify the xx suffix of this SYS1.PARMLIB member for each channel initiator. This causes the channel initiator to issue the required z/OS command SET APPC= xx to activate its LUNAME.

Define your ARM policy so that it restarts the channel initiator only if it fails while its z/OS image stays up; the user ID associated with the XCFAS address space must be authorized to issue the IBM MQ command START CHINIT. Do not restart the channel initiator automatically if its z/OS image also fails, instead use commands in the CSQINP2 and CSQINPX data sets to start the channel initiator and listeners.

## Recovering units of work manually

You can manually recover units of work CICS, IMS, RRS, or other queue managers in a queue sharing group. You can use queue manager commands to display the status of the units of work associated with each connection to the queue manager.

This topic contains information about the following subjects:

- [“Displaying connections and threads” on page 564](#)
- [“Recovering CICS units of recovery manually” on page 564](#)
- [“Recovering IMS units of recovery manually” on page 568](#)
- [“Recovering RRS units of recovery manually” on page 569](#)
- [“Recovering units of recovery on another queue manager in the queue sharing group” on page 570](#)

## Displaying connections and threads

You can use the `DISPLAY CONN` command to get information about connections to queue managers and their associated units of work. You can display active units of work to see what is currently happening, or to see what needs to be terminated to allow the queue manager to shut down, and you can display unresolved units of work to help with recovery.

### Active units of work

To display only active units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ ACTIVE)
```

### Unresolved units of work

An unresolved unit of work, also known as an "in-doubt thread", is one that is in the second pass of the two-phase commit operation. Resources are held in IBM MQ on its behalf. To display unresolved units of work, use

```
DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

External intervention is needed to resolve the status of unresolved units of work. This might only involve starting the recovery coordinator ( CICS, IMS, or RRS) or might involve more, as described in the following sections.

## **Recovering CICS units of recovery manually**

Use this topic to understand what happens when the CICS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

### What happens when the CICS adapter restarts

Whenever a connection is broken, the adapter has to go through a *restart phase* during the *reconnect process*. The restart phase resynchronizes resources. Resynchronization between CICS and IBM MQ enables in-doubt units of work to be identified and resolved.

Resynchronization can be caused by:

- An explicit request from the distributed queuing component
- An implicit request when a connection is made to IBM MQ

If the resynchronization is caused by connecting to IBM MQ, the sequence of events is:

1. The connection process retrieves a list of in-doubt units of work (UOW) IDs from IBM MQ.
2. The UOW IDs are displayed on the console in CSQC313I messages.
3. The UOW IDs are passed to CICS.
4. CICS initiates a resynchronization task (CRSY) for each in-doubt UOW ID.

5. The result of the task for each in-doubt UOW is displayed on the console.

You need to check the messages that are displayed during the connect process:

#### **CSQC313I**

Shows that a UOW is in doubt.

#### **CSQC400I**

Identifies the UOW and is followed by one of these messages:

- CSQC402I or CSQC403I shows that the UOW was resolved successfully (committed or backed out).
- CSQC404E, CSQC405E, CSQC406E, or CSQC407E shows that the UOW was not resolved.

#### **CSQC409I**

Shows that all UOWs were resolved successfully.

#### **CSQC408I**

Shows that not all UOWs were resolved successfully.

#### **CSQC314I**

Warns that UOW IDs highlighted with a \* are not resolved automatically. These UOWs must be resolved explicitly by the distributed queuing component when it is restarted.

Figure 37 on page 565 shows an example set of restart messages displayed on the z/OS console.

```
CSQ9022I +CSQ1 CSQYASCP ' START QMGR' NORMAL COMPLETION
+CSQC323I VICIC1 CSQCQCON CONNECT received from TERMID=PB62 TRANID=CKCN
+CSQC303I VICIC1 CSQCQCON CSQCSERV loaded. Entry point is 850E8918
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6F0E2178D25 is in doubt
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6F055B2AC25 is in doubt
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6EFFD60D425 is in doubt
+CSQC313I VICIC1 CSQCQCON UOWID=VICIC1.A6E5A6F07AB56D22 is in doubt
+CSQC307I VICIC1 CSQCQCON Successful connection to subsystem VC2
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAD18) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BAA10) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008BA708) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAE88) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CAB80) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA878) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA570) connect
successful
+CSQC472I VICIC1 CSQCSERV Server subtask (TCB address=008CA268) connect
successful
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F0E2178D25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F055B2AC25
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6F07AB56D22
+CSQC403I VICIC1 CSQCTRUE Resolved BACKOUT for
+CSQC400I VICIC1 CSQCTRUE UOWID=VICIC1.A6E5A6EFFD60D425
+CSQC409I VICIC1 CSQCTRUE Resynchronization completed successfully
```

Figure 37. Example restart messages

The total number of CSQC313I messages should equal the total number of CSQC402I plus CSQC403I messages. If the totals are not equal, there are UOWs that the connection process cannot resolve. Those UOWs that cannot be resolved are caused by problems with CICS (for example, a cold start) or with IBM MQ, or by distributing queuing. When these problems have been fixed, you can initiate another resynchronization by disconnecting and then reconnecting.

Alternatively, you can resolve each outstanding UOW yourself using the RESOLVE INDOUBT command and the UOW ID shown in message CSQC400I. You must then initiate a disconnect and a connect to clean

up the *unit of recovery descriptors* in CICS. You need to know the correct outcome of the UOW to resolve UOWs manually.

All messages that are associated with unresolved UOWs are locked by IBM MQ and no Batch, TSO, or CICS task can access them.

If CICS fails and an emergency restart is necessary, *do not* vary the GENERIC APPLID of the CICS system. If you do and then reconnect to IBM MQ, data integrity with IBM MQ cannot be guaranteed. This is because IBM MQ treats the new instance of CICS as a different CICS (because the APPLID is different). In-doubt resolution is then based on the wrong CICS log.

## How to resolve CICS units of recovery manually

If the adapter ends abnormally, CICS and IBM MQ build in-doubt lists either dynamically or during restart, depending on which subsystem caused the abend.

**Note:** If you use the DFH\$INDB sample program to show units of work, you might find that it does not always show IBM MQ UOWs correctly.

When CICS connects to IBM MQ, there might be one or more units of recovery that have not been resolved.

One of the following messages is sent to the console:

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E
- CSQC408I

For details of what these messages mean, see the [CICS adapter and Bridge messages](#) messages.

CICS retains details of units of recovery that were not resolved during connection startup. An entry is purged when it no longer appears on the list presented by IBM MQ.

Any units of recovery that CICS cannot resolve must be resolved manually using IBM MQ commands. This manual procedure is rarely used within an installation, because it is required only where operational errors or software problems have prevented automatic resolution. *Any inconsistencies found during in-doubt resolution must be investigated.*

To resolve the units of recovery:

1. Obtain a list of the units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN( * ) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```

CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN (BC85772CBE3E0001)
EXTCONN (C3E2D8C3C7D9F0F940404040404040)
TYPE (CONN)
CONNOPTS (
MQCNO_STANDARD_BINDING
)
UOWLOGDA (2005-02-04)
UOWLOGTI (10.17.44)
UOWSTDA (2005-02-04)
UOWSTTI (10.17.44)
UOWSTATE (UNRESOLVED)
NID (IYRCSQ1 .BC8571519B60222D)
EXTURID (BC8571519B60222D)
QMURID (0000002BDA50)
URTYPE (CICS)
USERID (MQTEST)
APPLTAG (IYRCSQ1)
ASID (0000)
APPLTYPE (CICS)
TRANSID (GP02)
TASKNO (0000096)
END CONN DETAILS

```

For CICS connections, the NID consists of the CICS applid and a unique number provided by CICS at the time the syncpoint log entries are written. This unique number is stored in records written to both the CICS system log and the IBM MQ log at syncpoint processing time. This value is referred to in CICS as the *recovery token*.

2. Scan the CICS log for entries related to a particular unit of recovery.

Look for a PREPARE record for the task-related installation where the recovery token field (JCSRMTKN) equals the value obtained from the network ID. The network ID is supplied by IBM MQ in the DISPLAY CONN command output.

The PREPARE record in the CICS log for the units of recovery provides the CICS task number. All other entries on the log for this CICS task can be located using this number.

You can use the CICS journal print utility DFHJUP when scanning the log. For details of using this program, see the *CICS Operations and Utilities Guide*.

3. Scan the IBM MQ log for records with the NID related to a particular unit of recovery. Then use the URID from this record to obtain the rest of the log records for this unit of recovery.

When scanning the IBM MQ log, note that the IBM MQ startup message CSQJ001I provides the start RBA for this session.

The print log records program (CSQ1LOGP) can be used for that purpose.

4. If you need to, do in-doubt resolution in IBM MQ.

IBM MQ can be directed to take the recovery action for a unit of recovery using an IBM MQ [RESOLVE INDOUBT](#) command.

To recover all threads associated with a specific *connection-name*, use the NID(\*) option.

The command produces one of the following messages showing whether the thread is committed or backed out:

```

CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id ABORT SCHEDULED

```

When performing in-doubt resolution, CICS and the adapter are not aware of the commands to IBM MQ to commit or back out units of recovery, because only IBM MQ resources are affected. However, CICS keeps details about the in-doubt threads that could not be resolved by IBM MQ. This information is purged

either when the list presented is empty, or when the list does not include a unit of recovery of which CICS has details.

## **Recovering IMS units of recovery manually**

Use this topic to understand what happens when the IMS adapter restarts, and then explains how to deal with any unresolved units of recovery that arise.

### **What happens when the IMS adapter restarts**

Whenever the connection to IBM MQ is restarted, either following a queue manager restart or an IMS / START SUBSYS command, IMS initiates the following resynchronization process:

1. IMS presents the list of unit of work (UOW) IDs that it believes are in doubt to the IBM MQ IMS adapter one at a time with a resolution parameter of Commit or Backout.
2. The IMS adapter passes the resolution request to IBM MQ and reports the result back to IMS.
3. Having processed all the IMS resolution requests, the IMS adapter gets from IBM MQ a list of all UOWs that IBM MQ still holds in doubt that were initiated by the IMS system. These are reported to the IMS master terminal in message CSQQ008I.

**Note:** While a UOW is in doubt, any associated IBM MQ message is locked by IBM MQ and is not available to any application.

### **How to resolve IMS units of recovery manually**

When IMS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved.

If IBM MQ has in-doubt units of recovery that IMS did not resolve, the following message is issued at the IMS master terminal:

```
CSQQ008I nn units of recovery are still in doubt in queue manager qmgr-name
```

If this message is issued, IMS was either cold-started or it was started with an incomplete log tape. This message can also be issued if IBM MQ or IMS terminates abnormally because of a software error or other subsystem failure.

After receiving the CSQQ008I message:

- The connection remains active.
- IMS applications can still access IBM MQ resources.
- Some IBM MQ resources remain locked out.

If the in-doubt thread is not resolved, IMS message queues can start to build up. If the IMS queues fill to capacity, IMS terminates. You must be aware of this potential difficulty, and you must monitor IMS until the in-doubt units of recovery are fully resolved.

### **Recovery procedure**

Use the following procedure to recover the IMS units of work:

1. Force the IMS log closed, using /SWI OLDS, and then archive the IMS log. Use the utility, DFSERA10, to print the records from the previous IMS log tape. Type X ' 3730 ' log records indicate a phase-2 commit request and type X ' 38 ' log records indicate an abort request. Record the requested action for the last transaction in each dependent region.
2. Run the DL/I batch job to back out each PSB involved that has not reached a commit point. The process might take some time because transactions are still being processed. It might also lock



up a number of records, which could affect the rest of the processing and the rest of the message queues.

3. Produce a list of the in-doubt units of recovery from IBM MQ using the following command:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

You receive the following message:

```
CSQM201I +CSQ1 CSQMDRTC DISPLAY CONN DETAILS
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3E2C5C3F240404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2005-02-15)
UOWLOGTI(16.39.43)
UOWSTDA(2005-02-15)
UOWSTTI(16.39.43)
UOWSTATE(UNRESOLVED)
NID(IM8F .BC45A794D3810344)
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0000)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

For IMS, the NID consists of the IMS connection name and a unique number provided by IMS. The value is referred to in IMS as the *recovery token*. For more information, see the [IMS documentation](#).

4. Compare the NIDs (IMSID plus OASN in hexadecimal) displayed in the DISPLAY THREAD messages with the OASNs (4 bytes decimal) shown in the DFSERA10 output. Decide whether to commit or back out.
5. Perform in-doubt resolution in IBM MQ with the [RESOLVE INDOUBT](#) command, as follows:

```
RESOLVE INDOUBT( connection-name )
ACTION(COMMIT|BACKOUT)
NID( network-id )
```

To recover all threads associated with *connection-name*, use the NID(\*) option. The command results in one of the following messages to indicate whether the thread is committed or backed out:

```
CSQV414I THREAD network-id COMMIT SCHEDULED
CSQV415I THREAD network-id BACKOUT SCHEDULED
```

When performing in-doubt resolution, IMS and the adapter are not aware of the commands to IBM MQ to commit or back out in-doubt units of recovery because only IBM MQ resources are affected.

## **Recovering RRS units of recovery manually**

Use this topic to understand the how to determine if there are in-doubt RRS units of recovery, and how to manually resolve those units of recovery.

When RRS connects to IBM MQ, IBM MQ might have one, or more in-doubt units of recovery that have not been resolved. If IBM MQ has in-doubt units of recovery that RRS did not resolve, one of the following messages is issued at the z/OS console:

- CSQ3011I
- CSQ3013I

- CSQ3014I
- CSQ3016I

Both IBM MQ and RRS provide tools to display information about in-doubt units of recovery, and techniques for manually resolving them.

In IBM MQ, use the DISPLAY CONN command to display information about in-doubt IBM MQ threads. The output from the command includes RRS unit of recovery IDs for those IBM MQ threads that have RRS as a coordinator. This can be used to determine the outcome of the unit of recovery.

Use the RESOLVE INDOUBT command to resolve the IBM MQ in-doubt thread manually. This command can be used to either commit or back out the unit of recovery after you have determined what the correct decision is.

### **Recovering units of recovery on another queue manager in the queue sharing group**

Use this topic to identify, and manually recover units of recovery on other queue managers in a queue sharing group.

If a queue manager that is a member of a queue sharing group fails and cannot be restarted, other queue managers in the group can perform peer recovery, and take over from it. However, the queue manager might have in-doubt units of recovery that cannot be resolved by peer recovery because the final disposition of that unit of recovery is known only to the failed queue manager. These units of recovery are resolved when the queue manager is eventually restarted, but until then, they remain in doubt.

This means that certain resources (for example, messages) might be locked, making them unavailable to other queue managers in the group. In this situation, you can use the DISPLAY THREAD command to display these units of work on the inactive queue manager. If you want to resolve these units of recovery manually to make the messages available to other queue managers in the group, you can use the RESOLVE INDOUBT command.

When you issue the DISPLAY THREAD command to display units of recovery that are in doubt, you can use the QMNAME keyword to specify the name of the inactive queue manager. For example, if you issue the following command:

```
+CSQ1 DISPLAY THREAD(*) TYPE(INDOUBT) QMNAME(QM01)
```

You receive the following messages:

```
CSQV436I +CSQ1 INDOUBT THREADS FOR QM01 -
NAME   THREAD-XREF   URID NID
USER1  0000000000000000000000000000 CSQ:0001.0
USER2  0000000000000000000000000000 CSQ:0002.0
DISPLAY THREAD REPORT COMPLETE
```

If the queue manager specified is active, IBM MQ does not return information about in-doubt threads, but issues the following message:

```
CSQV435I CANNOT USE QMNAME KEYWORD, QM01 IS ACTIVE
```

Use the IBM MQ command RESOLVE INDOUBT to resolve the in-doubt threads manually. Use the QMNAME keyword to specify the name of the inactive queue manager in the command.

This command can be used to commit or back out the unit of recovery. The command resolves the shared portion of the unit of recovery only; any local messages are unaffected and remain locked until the queue manager restarts, or reconnects to CICS, IMS, or RRS batch.

## **z/OS IBM MQ and IMS**

IBM MQ provides two components to interface with IMS, the IBM MQ - IMS adapter, and the IBM MQ - IMS bridge. These components are commonly called the IMS adapter, and the IMS bridge.

### **z/OS Operating the IMS adapter**

Use this topic to understand how to operate the IMS adapter, which connects IBM MQ to IMS systems.

**Note:** The IMS adapter does not incorporate any operations and control panels.

This topic contains the following sections:

- [“Controlling IMS connections” on page 571](#)
- [“Connecting from the IMS control region” on page 571](#)
- [“Displaying in-doubt units of recovery” on page 573](#)
- [“Controlling IMS dependent region connections” on page 575](#)
- [“Disconnecting from IMS” on page 577](#)
- [“Controlling the IMS trigger monitor” on page 578](#)

### **z/OS Controlling IMS connections**

Use this topic to understand the IMS operator commands which control and monitor the connection to IBM MQ.

IMS provides the following operator commands to control and monitor the connection to IBM MQ:

#### **/CHANGE SUBSYS**

Deletes an in-doubt unit of recovery from IMS.

#### **/DISPLAY OASN SUBSYS**

Displays outstanding recovery elements.

#### **/DISPLAY SUBSYS**

Displays connection status and thread activity.

#### **/START SUBSYS**

Connects the IMS control region to a queue manager.

#### **/STOP SUBSYS**

Disconnects IMS from a queue manager.

#### **/TRACE**

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA® Operator's Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

### **z/OS Connecting from the IMS control region**

Use this topic to understand the mechanisms available to connect from IMS to IBM MQ.

IMS makes one connection from its control region to each queue manager that uses IMS. IMS must be enabled to make the connection in one of these ways:

- Automatically during either:
  - A cold start initialization.
  - A warm start of IMS, if the IBM MQ connection was active when IMS was shut down.
- In response to the IMS command:

```
/START SUBSYS sysid
```

where *sysid* is the queue manager name.

The command can be issued regardless of whether the queue manager is active.

The connection is not made until the first IBM MQ API call to the queue manager is made. Until that time, the IMS command /DIS SUBSYS shows the status as 'NOT CONN'.

The order in which you start IMS and the queue manager is not significant.

IMS cannot re-enable the connection to the queue manager automatically if the queue manager is stopped with a STOP QMGR command, the IMS command /STOP SUBSYS, or an abnormal end. Therefore, you must make the connection by using the IMS command /START SUBSYS.

If an IMS command is seen in the queue manager console log similar to this:

```
MODIFY IMS*,SS*
```

check the IMS master log and ensure that IBM MQ has RACF authority to issue IMS Adapter MODIFY commands.

## Initializing the adapter and connecting to the queue manager

The adapter is a set of modules loaded into the IMS control and dependent regions, using the IMS external Subsystem Attach Facility.

This procedure initializes the adapter and connects to the queue manager:

1. Read the subsystem member (SSM) from IMS.PROCLIB. The SSM chosen is an IMS EXEC parameter. There is one entry in the member for each queue manager to which IMS can connect. Each entry contains control information about an IBM MQ adapter.

2. Load the IMS adapter.

**Note:** IMS loads one copy of the adapter modules for each IBM MQ instance that is defined in the SSM member.

3. Attach the external subsystem task for IBM MQ.
4. Run the adapter with the CTL EXEC parameter (IMSID) as the connection name.

The process is the same whether the connection is part of initialization or a result of the IMS command /START SUBSYS.

If the queue manager is active when IMS tries to make the connection, the following messages are sent:

- to the z/OS console:

```
DFS3613I ESS TCB INITIALIZATION COMPLETE
```

- to the IMS master terminal:

```
CSQQ000I IMS/TM imsid connected to queue manager ssnm
```

When IMS tries to make the connection and *the queue manager is not active*, the following messages are sent to the IMS master terminal each time an application makes an MQI call:

```
CSQQ001I IMS/TM imsid not connected to queue manager ssnm.  
Notify message accepted  
DFS3607I MQM1 SUBSYSTEM ID EXIT FAILURE, FC = 0286, RC = 08,  
JOBNAME = IMSEMPR1
```

If you get DFS3607I messages when you start the connection to IMS or on system startup, this indicates that the queue manager is not available. To prevent a large number of messages being generated, you must do one of the following:

1. Start the relevant queue manager.
2. Issue the IMS command:

```
/STOP SUBSYS
```

so that IMS does not expect to connect to the queue manager.

If you do neither, a DFS3607I message and the associated CSQQ001I message are issued each time a job is scheduled in the region and each time a connection request to the queue manager is made by an application.

## Thread attachment

In an MPP or IFP region, IMS makes a thread connection when the first application program is scheduled into that region, even if that application program does not make an IBM MQ call. In a BMP region, the thread connection is made when the application makes its first IBM MQ call (MQCONN or MQCONNX). This thread is retained for the duration of the region or until the connection is stopped.

For both the message driven and non-message driven regions, the recovery thread cross-reference identifier, *Thread-xref*, associated with the thread is:

```
PSTid + PSBname
```

where:

### **PSTid**

Partition specification table region identifier

### **PSBname**

Program specification block name

You can use connection IDs as unique identifiers in IBM MQ commands, in which case IBM MQ automatically inserts these IDs into any operator message that it generates.

## **Displaying in-doubt units of recovery**

You can display in-doubt units of recovery and attempt to recover them.

The operational steps used to list and recover in-doubt units of recovery in this topic are for relatively simple cases only. If the queue manager ends abnormally while connected to IMS, IMS might commit or back out work without IBM MQ being aware of it. When the queue manager restarts, that work is termed *in doubt*. A decision must be made about the status of the work.

To display a list of in-doubt units of recovery, issue the command:

```
+CSQ1 DISPLAY CONN(*) WHERE(UOWSTATE EQ UNRESOLVED)
```

IBM MQ responds with a message like the following:

```
CSQM201I +CSQ1 CSQMDRTC DIS CONN DETAILS
CONN(BC0F6125F5A30001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-11-02)
UOWLOGTI(12.27.58)
UOWSTDA(2004-11-02)
UOWSTTI(12.27.58)
UOWSTATE(UNRESOLVED)
NID(CSQ1CHIN.BC0F5F1C86FC0766)
EXTURID(00000000000001F000000007472616E5F6964547565204E6F762020...)
QMURID(0000000026232)
URTYPE(XA)
USERID( )
APPLTAG(CSQ1CHIN)
ASID(0000)
APPLTYPE(CHINIT)
CHANNEL( )
CONNNAME( )
END CONN DETAILS
```

For an explanation of the attributes in this message, see the description of the [DISPLAY CONN](#) command.

## Recovering in-doubt units of recovery

To recover in-doubt units of recovery, issue this command:

```
+CSQ1 RESOLVE INDOUBT( connection-name ) ACTION(COMMIT|BACKOUT)
NID( net-node.number )
```

where:

### ***connection-name***

The IMS system ID.

### **ACTION**

Indicates whether to commit (COMMIT) or back out (BACKOUT) this unit of recovery.

### ***net-node.number***

The associated net-node.number.

When you have issued the RESOLVE INDOUBT command, one of the following messages is displayed:

```
CSQV414I +CSQ1 THREAD network-id COMMIT SCHEDULED
CSQV415I +CSQ1 THREAD network-id BACKOUT SCHEDULED
```

## Resolving residual recovery entries

At given times, IMS builds a list of residual recovery entries (RREs). RREs are units of recovery about which IBM MQ might be in doubt. They arise in several situations:

- If the queue manager is not active, IMS has RREs that cannot be resolved until the queue manager is active. These RREs are not a problem.
- If the queue manager is active and connected to IMS, and if IMS backs out the work that IBM MQ has committed, the IMS adapter issues message CSQQ010E. If the data in the two systems must be consistent, there is a problem. For information about resolving this problem, see [“Recovering IMS units of recovery manually”](#) on page 568.
- If the queue manager is active and connected to IMS, there might still be RREs even though no messages have informed you of this problem. After the IBM MQ connection to IMS has been established, you can issue the following IMS command to find out if there is a problem:

```
/DISPLAY OASN SUBSYS sysid
```

To purge the RRE, issue one of the following IMS commands:

```
/CHANGE SUBSYS sysid RESET  
/CHANGE SUBSYS sysid RESET OASN nnnn
```

where *nnnn* is the originating application sequence number listed in response to your +CSQ1 DISPLAY command. This is the schedule number of the program instance, giving its place in the sequence of invocations of that program since the last IMS cold start. IMS cannot have two in-doubt units of recovery with the same schedule number.

These commands reset the status of IMS ; they do not result in any communication with IBM MQ.

## **Controlling IMS dependent region connections**

You can control, monitor, and, when necessary, terminate connections between IMS and IBM MQ.

Controlling IMS dependent region connections involves the following activities:

- [Connecting from dependent regions](#)
- [Region error options](#)
- [Monitoring the activity on connections](#)
- [Disconnecting from dependent regions](#)

## Connecting from dependent regions

The IMS adapter used in the control region is also loaded into dependent regions. A connection is made from each dependent region to IBM MQ. This connection is used to coordinate the commitment of IBM MQ and IMS work. To initialize and make the connection, IMS does the following:

1. Reads the subsystem member (SSM) from IMS.PROCLIB.

A subsystem member can be specified on the dependent region EXEC parameter. If it is not specified, the control region SSM is used. If the region is never likely to connect to IBM MQ, to avoid loading the adapter, specify a member with no entries.

2. Loads the IBM MQ adapter.

For a batch message program, the load is not done until the application issues its first messaging command. At that time, IMS tries to make the connection.

For a message-processing program region or IMS fast-path region, the attempt is made when the region is initialized.

## Region error options

If the queue manager is not active, or if resources are not available when the first messaging command is sent from application programs, the action taken depends on the error option specified on the SSM entry. The options are:

### R

The appropriate return code is sent to the application.

### Q

The application ends abnormally with abend code U3051. The input message is re-queued.

### A

The application ends abnormally with abend code U3047. The input message is discarded.

## Monitoring the activity on connections

A thread is established from a dependent region when an application makes its first successful IBM MQ request. You can display information about connections and the applications currently using them by issuing the following command from IBM MQ:

```
+CSQ1 DISPLAY CONN(*) ALL
```

The command produces a message like the following:

```
CONN(BC45A794C4290001)
EXTCONN(C3E2D8C3C3E2D8F140404040404040)
TYPE(CONN)
CONNOPTS(
MQCNO_STANDARD_BINDING
)
UOWLOGDA(2004-12-15)
UOWLOGTI(16.39.43)
UOWSTDA(2004-12-15)
UOWSTTI(16.39.43)
UOWSTATE(ACTIVE)
NID( )
EXTURID(
0000052900000000
)
QMURID(00000354B76E)
URTYPE(IMS)
USERID(STCPI)
APPLTAG(IM8F)
ASID(0049)
APPLTYPE(IMS)
PSTID(0004)
PSBNAME(GP01MPP)
```

For the control region, *thread-xref* is the special value CONTROL. For dependent regions, it is the PSTid concatenated with the PSBname. *auth-id* is either the user field from the job card, or the ID from the z/OS started procedures table.

For an explanation of the displayed list, see the description of message CSQV402I in the [Messages, codes achèvement et codes anomalie IBM MQ for z/OS documentation](#).



IMS provides a display command to monitor the connection to IBM MQ. It shows which program is active on each dependent region connection, the LTERM user name, and the control region connection status. The command is:

```
/DISPLAY SUBSYS name
```

The status of the connection between IMS and IBM MQ is shown as one of:

```
CONNECTED  
NOT CONNECTED  
CONNECT IN PROGRESS  
STOPPED  
STOP IN PROGRESS  
INVALID SUBSYSTEM NAME= name  
SUBSYSTEM name NOT DEFINED BUT RECOVERY OUTSTANDING
```

The thread status from each dependent region is one of the following:

```
CONN  
CONN, ACTIVE (includes LTERM of user)
```

## Disconnecting from dependent regions

To change values in the SSM member of IMS.PROCLIB, you disconnect a dependent region. To do this, you must:

1. Issue the IMS command:

```
/STOP REGION
```

2. Update the SSM member.
3. Issue the IMS command:

```
/START REGION
```

## **Disconnecting from IMS**

The connection is ended when either IMS or the queue manager terminates. Alternatively, the IMS master terminal operator can explicitly break the connection.

To terminate the connection between IMS and IBM MQ, use the following IMS command:

```
/STOP SUBSYS sysid
```

The command sends the following message to the terminal that issued it, typically the master terminal operator (MTO):

```
DFS058I STOP COMMAND IN PROGRESS
```

The IMS command:

```
/START SUBSYS sysid
```

is required to reestablish the connection.

**Note:** The IMS command /STOP SUBSYS is not completed if an IMS trigger monitor is running.

## **Controlling the IMS trigger monitor**

You can use the CSQQTRMN transaction to stop, and start the IMS trigger monitor.

The IMS trigger monitor (the CSQQTRMN transaction) is described in the [Setting up the IMS trigger monitor](#).

To control the IMS trigger monitor see:

- [Starting CSQQTRMN](#)
- [Stopping CSQQTRMN](#)

## Starting CSQQTRMN

1. Start a batch-oriented BMP that runs the program CSQQTRMN for each initiation queue you want to monitor.
2. Modify your batch JCL to add a DDname of CSQQUT1 that points to a data set containing the following information:

```
QMGRNAME=q_manager_name    Comment: queue manager name
INITQUEUEUENAME=init_q_name  Comment: initiation queue name
LTERM=lterm                  Comment: LTERM to remove error messages
CONSOLEMESSAGES=YES         Comment: Send error messages to console
```

where:

q_manager_name	The name of the queue manager (if this is blank, the default nominated in CSQQDEFV is assumed)
init_q_name	The name of the initiation queue to be monitored
lterm	The IMS LTERM name for the destination of error messages (if this is blank, the default value is MASTER).
CONSOLEMESSAGES= YES	Requests that messages sent to the nominated IMS LTERM are also sent to the z/OS console. If this parameter is omitted or misspelled, the default is NOT to send messages to the console.

3. Add a DD name of CSQQUT2 if you want a printed report of the processing of CSQQUT1 input.

### **Note:**

1. The data set CSQQUT1 is defined with LRECL=80. Other DCB information is taken from the data set. The DCB for data set CSQQUT2 is RECFM=VBA and LRECL=125.

2. You can put only one keyword on each record. The keyword value is delimited by the first blank following the keyword; this means that you can include comments. An asterisk in column 1 means that the whole input record is a comment.
3. If you misspell either of the QMGRNAME or LTERM keywords, CSQQTRMN uses the default for that keyword.
4. Ensure that the subsystem is started in IMS (by the /START SUBSYS command) before submitting the trigger monitor BMP job. If it is not started, your trigger monitor job terminates with abend code U3042.

## Stopping CSQQTRMN

Once started, CSQQTRMN runs until either the connection between IBM MQ and IMS is broken due to one of the following events:

- the queue manager ending
- IMS ending

or a z/OS STOP **jobname** command is entered.

## Controlling the IMS bridge

Use this topic to understand the IMS commands that you can use to control the IMS bridge.

There are no IBM MQ commands to control the IBM MQ-IMS bridge. However, you can stop messages being delivered to IMS in the following ways:

- For non-shared queues, by using the ALTER QLOCAL(xxx) GET(DISABLED) command for all bridge queues.
- For clustered queues, by using the SUSPEND QMGR CLUSTER(xxx) command. This is effective only when another queue manager is also hosting the clustered bridge queue.
- For clustered queues, by using the SUSPEND QMGR FACILITY(IMSBRIDGE) command. No further messages are sent to IMS, but the responses for any outstanding transactions are received from IMS.

To start sending messages to IMS again, issue the RESUME QMGR FACILITY(IMSBRIDGE) command.

You can also use the MQSC command DISPLAY SYSTEM to display whether the bridge is suspended.

See [MQSC commands](#) for details of these commands.

For further information see:

- [“Starting and stopping the IMS bridge” on page 579](#)
- [“Controlling IMS connections” on page 580](#)
- [Controlling bridge queues](#)
- [“Resynchronizing the IMS bridge” on page 581](#)
- [Working with tpipe names](#)
- [Deleting messages from IMS](#)
- [Deleting tpipes](#)
- [“IMS Transaction Expiration” on page 583](#)

## Starting and stopping the IMS bridge

Start the IBM MQ bridge by starting OTMA. Either use the IMS command:

```
/START OTMA
```

or start it automatically by specifying OTMA=YES in the IMS system parameters. If OTMA is already started, the bridge starts automatically when queue manager startup has completed. An IBM MQ event message is produced when OTMA is started.

Use the IMS command:

```
/STOP OTMA
```

to stop OTMA communication. When this command is issued, an IBM MQ event message is produced.

## Controlling IMS connections

IMS provides these operator commands to control and monitor the connection to IBM MQ:

### **/DEQUEUE TMEMBER *tmember* TPIPE *tpipe***

Removes messages from a Tpipe. Specify PURGE to remove all messages or PURGE1 to remove the first message only.

### **/DISPLAY OTMA**

Displays summary information about the OTMA server and clients, and client status.

### **/DISPLAY TMEMBER *name***

Displays information about an OTMA client.

### **/DISPLAY TRACE TMEMBER *name***

Displays information about what is being traced.

### **/SECURE OTMA**

Sets security options.

### **/START OTMA**

Enables communications through OTMA.

### **/START TMEMBER *tmember* TPIPE *tpipe***

Starts the named Tpipe.

### **/STOP OTMA**

Stops communications through OTMA.

### **/STOP TMEMBER *tmember* TPIPE *tpipe***

Stops the named Tpipe.

### **/TRACE**

Controls the IMS trace.

For more information about these commands, see the *IMS/ESA Operators Reference* manual for the level of IMS that you are using.

IMS command responses are sent to the terminal from which the command was issued. Authorization to issue IMS commands is based on IMS security.

## Controlling bridge queues

To stop communicating with the queue manager with XCF member name *tmember* through the bridge, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE ALL
```

To resume communication, issue the following IMS command:

```
/START TMEMBER tmember TPIPE ALL
```

The Tpipes for a queue can be displayed using the MQ DISPLAY QUEUE command.

To stop communication with the queue manager on a single Tpipe, issue the following IMS command:

```
/STOP TMEMBER tmember TPIPE tpipe
```

One or two Tpipes are created for each active bridge queue, so issuing this command stops communication with the IBM MQ queue. To resume communication, use the following IMS command :

```
/START TMEMBER tmember TPIPE tpipe
```

Alternatively, you can alter the attributes of the IBM MQ queue to make it get inhibited.

## Resynchronizing the IMS bridge

The IMS bridge is automatically restarted whenever the queue manager, IMS, or OTMA are restarted.

The first task undertaken by the IMS bridge is to resynchronize with IMS. This involves IBM MQ and IMS checking sequence numbers on every synchronized Tpipe. A synchronized Tpipe is used when persistent messages are sent to IMS from an IBM MQ - IMS bridge queue using commit mode zero (commit-then-send).

If the bridge cannot resynchronize with IMS, the IMS sense code is returned in message CSQ203E and the connection to OTMA is stopped. If the bridge cannot resynchronize with an individual IMS Tpipe, the IMS sense code is returned in message CSQ2025E and the Tpipe is stopped. If a Tpipe has been cold started, the recoverable sequence numbers are automatically reset to 1.

If the bridge discovers mismatched sequence numbers when resynchronizing with a Tpipe, message CSQ2020E is issued. Use the IBM MQ command RESET TPIPE to initiate resynchronization with the IMS Tpipe. You need to provide the XCF group and member name, and the name of the Tpipe; this information is provided by the message.

You can also specify:

- A new recoverable sequence number to be set in the Tpipe for messages sent by IBM MQ, and to be set as the partner's receive sequence number. If you do not specify this, the partner's receive sequence number is set to the current IBM MQ send sequence number.
- A new recoverable sequence number to be set in the Tpipe for messages received by IBM MQ, and to be set as the partner's send sequence number. If you do not specify this, the partner's send sequence number is set to the current IBM MQ receive sequence number.

If there is an unresolved unit of recovery associated with the Tpipe, this is also notified in the message. Use the IBM MQ command RESET TPIPE to specify whether to commit the unit of recovery, or back it out. If you commit the unit of recovery, the batch of messages has already been sent to IMS, and is deleted from the bridge queue. If you back the unit of recovery out, the messages are returned to the bridge queue, to be later sent to IMS.

Commit mode 1 (send-then-commit) Tpipes are not synchronized.

### Considerations for Commit mode 1 transactions

In IMS, commit mode 1 (CM1) transactions send their output replies before sync point.

A CM1 transaction might not be able to send its reply, for example because:

- The Tpipe on which the reply is to be sent is stopped
- OTMA is stopped
- The OTMA client (that is, the queue manager) has gone away
- The reply-to queue and dead-letter queue are unavailable

For these reasons, the IMS application sending the message pseudo-abends with code U0119. The IMS transaction and program are not stopped in this case.

These reasons often prevent messages being sent into IMS, as well as replies being delivered from IMS. A U0119 abend can occur if:

- The Tpipe, OTMA, or the queue manager is stopped while the message is in IMS
- IMS replies on a different Tpipe to the incoming message, and that Tpipe is stopped
- IMS replies to a different OTMA client, and that client is unavailable.

Whenever a U0119 abend occurs, both the incoming message to IMS and the reply messages to IBM MQ are lost. If the output of a CMO transaction cannot be delivered for any of these reasons, it is queued on the Tpipe within IMS.

## Working with tpipe names

Many of the commands used to control the IBM MQ - IMS bridge require the *tpipe* name. Use this topic to understand how you can find further details of the tpipe name.

You need *tpipe* names for many of the commands that control the IBM MQ - IMS bridge. You can get the tpipe names from DISPLAY QUEUE command and note the following points:

- tpipe names are assigned when a local queue is defined
- a local queue is given two tpipe names, one for sync and one for non-sync
- tpipe names will not be known to IMS until after some communication between IMS and IBM MQ specific to that particular local queue takes place
- For a tpipe to be available for use by the IBM MQ - IMS bridge its associated queue must be assigned to a Storage Class that has the correct XCF group and member name fields completed

## Deleting messages from IMS

A message that is destined for IBM MQ through the IMS bridge can be deleted if the Tmember/Tpipe is stopped. To delete one message for the queue manager with XCF member name *tmember*, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE1
```

To delete all the messages on the Tpipe, issue the following IMS command:

```
/DEQUEUE TMEMBER tmember TPIPE tpipe PURGE
```

## Deleting tpipes

You cannot delete IMS tpipes yourself. They are deleted by IMS at the following times:

- Synchronized tpipes are deleted when IMS is cold started.

- Non-synchronized tpipes are deleted when IMS is restarted.

## IMS Transaction Expiration

An expiration time is associated with a transaction; any IBM MQ message can have an expiration time associated with it. The expiration interval is passed from the application, to IBM MQ, using the MQMD.Expiry field. The time is the duration of a message before it expires, expressed as a value in tenths of a second. An attempt to perform the MQGET of a message, later than it has expired, results in the message being removed from the queue and expiry processing performed. The expiration time decreases as a message flows between queue managers on an IBM MQ network. When an IMS message is passed across the IMS bridge to OTMA, the remaining message expiry time is passed to OTMA as a transaction expiration time.

If a transaction has an expiration time specified, OTMA expires the input transactions in three different places in IMS:

- input message receiving from XCF
- input message enqueueing time
- application GU time

No expiration is performed after the GU time.

The transaction EXPRTIME can be provided by:

- IMS transaction definition
- IMS OTMA message header
- IMS DFSINSX0 user exit
- IMS CREATE or UPDATE TRAN commands

IMS indicates that it has expired a transaction by abending a transaction with 0243, and issuing a message. The message issued is either DFS555I in the non-shared-queues environment, or DFS2224I in the shared-queues environment.

z/OS

## Operating Advanced Message Security on z/OS

The Advanced Message Security address space accepts commands using the z/OS MODIFY command.

### Procedure

- Modify Advanced Message Security on z/OS.

To enter commands for the Advanced Message Security (AMS) address space, use the z/OS MODIFY command.

For example:

```
F qmgrAMSM, cmd
```

where *qmgr* is the prefix of the started task name.

The following table describes the MODIFY commands that are accepted:

Table 29. Advanced Message Security address space MODIFY commands		
Command	Option	Description
DISPLAY		Display version information

Table 29. Advanced Message Security address space MODIFY commands (continued)		
Command	Option	Description
REFRESH	KEYRING POLICY ALL	Refresh the key ring certificates, security policies, or both.
SMFAUDIT	SUCCESS FAILURE ALL	Set whether SMF auditing is required when AMS successfully protects or unprotects messages, when AMS fails to protect or unprotect messages, or both.
SMFTYPE	0 - 255	Set the SMF record type to be generated when AMS protects or unprotects messages. To disable SMF auditing specify a record type of 0.

**Note:** To specify an option it must be separated by a comma. For example:

```
F qmgrAMSM,REFRESH KEYRING
F qmgrAMSM,SMFAUDIT ALL
F qmgrAMSM,SMFTYPE 180
```

- Refresh Advanced Message Security on z/OS.

Changes that are made effective by issuing the **REFRESH** command apply to applications that issue MQOPEN after the **REFRESH** command has completed. Existing applications that have a queue open, continue to use the options from when the application opened the queue. To use the new values, the application has to close and reopen the queue.

- Start and stop AMS on z/OS.

You do not need to enter a command to start or stop the Advanced Message Security address space. The AMS address space is started automatically when the queue manager is started if AMS has been enabled with the **SPLCAP** parameter of CSQ6SYSP, and is stopped when the queue manager is stopped.

## Administration de IBM MQ Internet Pass-Thru

Cette section explique comment administrer IBM MQ Internet Pass-Thru (MQIPT).



Configurez MQIPT en apportant des modifications au fichier de configuration `mqipt.conf`, comme décrit dans Configuration de IBM MQ Internet Pass-Thru. Pour administrer MQIPT, y compris l'actualisation de MQIPT afin d'appliquer les modifications de configuration sans redémarrer MQIPT, utilisez la commande **mqiptAdmin**. Pour plus d'informations sur l'administration de MQIPT à l'aide de la commande **mqiptAdmin**, voir «administration de MQIPT via la ligne de commande», à la page 587.

## Démarrage et arrêt de MQIPT

Vous pouvez démarrer MQIPT à partir de la ligne de commande, ou le faire démarrer automatiquement en même temps que le système. Vous pouvez arrêter MQIPT à l'aide de la commande **mqiptAdmin**.

### Démarrage de MQIPT à partir de la ligne de commande

MQIPT est installé dans un répertoire d'installation, par exemple :

-  **Windows** C : \MQIPT sur les systèmes Windows, avec des scripts exécutables dans C : \MQIPT\bin
-  **Linux** **AIX** /opt/mqipt sur les systèmes AIX and Linux, avec des scripts exécutables dans /opt/mqipt/bin



MQIPT utilise également un répertoire de base, qui contient le fichier de configuration `mqipt.conf` et tous les fichiers créés par MQIPT lorsqu'il est en cours d'exécution. Les sous-répertoires suivants du répertoire de base de MQIPT sont créés automatiquement lorsque MQIPT est appelé pour la première fois :

- Un répertoire `errors` dans lequel les fichiers First Failure Support Technology (FFST) et les fichiers de trace sont écrits.
- Un répertoire `logs` dans lequel est conservé le journal de connexion.

L'ID utilisateur sous lequel MQIPT est exécuté doit disposer des droits nécessaires pour créer ces répertoires, ou bien les répertoires doivent déjà exister et l'ID utilisateur doit disposer des droits nécessaires pour créer, lire et écrire dans les fichiers qu'ils contiennent. De plus, si vous utilisez une règle Java security manager, la règle de sécurité doit accorder les droits requis pour ces répertoires. Pour plus d'informations sur les paramètres de stratégie du gestionnaire de sécurité, voir [Java security manager](#).

Vous pouvez utiliser le répertoire d'installation en tant que répertoire de base. Si vous utilisez ce répertoire, vous devez vous assurer que l'ID utilisateur sous lequel MQIPT s'exécute dispose des droits appropriés et que la règle de gestionnaire de sécurité est correctement configurée.

Pour démarrer MQIPT, utilisez la commande `mqipt`, qui se trouve dans le répertoire `bin` du répertoire d'installation MQIPT. Par exemple, la commande suivante démarre une instance de MQIPT qui utilise le répertoire `C:\mqiptHome` comme répertoire de base:

```
mqipt C:\mqiptHome
```

Pour plus d'informations sur la commande `mqipt`, voir [mqipt \(start MQIPT\)](#).



Vous pouvez utiliser la commande `mqipt` pour spécifier un nom à attribuer à l'instance MQIPT en cours de démarrage. Le nom de l'instance MQIPT est utilisé pour administrer les instances locales de MQIPT avec la commande `mqiptAdmin` sans avoir à utiliser de port de commande. Si ce paramètre n'est pas spécifié, le nom du répertoire de base MQIPT est utilisé comme nom de l'instance MQIPT.

Les messages de console affichent l'état de MQIPT. Si une erreur se produit, voir [Traitement des incidents liés à IBM MQ Internet Pass-Thru](#). Les messages suivants sont un exemple de résultat lorsque MQIPT démarre correctement :

```
5724-H72 (C) Copyright IBM Corp. 2000, 2024. All Rights Reserved
MQCPI001 IBM MQ Internet Pass-Thru V9.2.0.0 starting
MQCPI004 Reading configuration information from mqipt.conf
MQCPI152 MQIPT name is C:\mqiptHome
MQCPI021 Password checking has been enabled on the command port
MQCPI144 MQ Advanced capabilities not enabled
MQCPI011 The path C:\mqiptHome\logs will be used to store the log files
MQCPI006 Route 1414 is starting and will forward messages to :
MQCPI034 ...examplehost(1414)
MQCPI035 ...using MQ protocols
MQCPI057 ...trace level 5 enabled
MQCPI078 Route 1414 ready for connection requests
```

## Démarrage automatique de MQIPT

Vous pouvez installer MQIPT en tant que service système qui démarre automatiquement lorsque le système est démarré. Utilisez la commande `mqiptService` pour installer et désinstaller le service MQIPT.

-  **Windows** Sur les systèmes Windows, la commande `mqiptService` installe MQIPT en tant que service Windows.
-  **Linux** **AIX** Sur les systèmes AIX and Linux, la commande `mqiptService` installe MQIPT en tant que service d'initialisation System V qui démarre à l'amorçage du système. Sur les systèmes Linux qui ne prennent pas en charge System V init, utilisez une autre méthode, telle que `systemd`, pour gérer MQIPT en tant que service.

Lorsque le service MQIPT est démarré, toutes les routes MQIPT actives démarrent. Lorsque le service est arrêté, tous les routes sont soumises à un arrêt immédiat.

Vous ne pouvez installer qu'un seul service MQIPT sur un système, même s'il existe plusieurs installations de MQIPT sur le système.

Pour plus d'informations sur la commande **mqiPTService**, voir [mqiPTService \(gestion du service MQIPT\)](#).

## Arrêt de MQIPT

Vous pouvez arrêter MQIPT à l'aide de la commande **mqiPTAdmin** avec le paramètre **-stop**.

Par exemple, la commande suivante arrête une instance de MQIPT portant le nom `mqiPT1` qui s'exécute localement sous le même ID utilisateur que la commande **mqiPTAdmin** :

```
mqiPTAdmin -stop -n ipt1
```

La commande **mqiPTAdmin** se connecte à l'instance active de MQIPT pour l'administrer à l'aide de l'une des méthodes suivantes:

- en se connectant à une instance locale de MQIPT sans utiliser le port de commande.
- en effectuant une connexion réseau à un port de commande.

L'arrêt à distance doit être activé en définissant la propriété **RemoteShutDown** sur `true` pour que la commande **mqiPTAdmin** puisse être utilisée pour arrêter MQIPT en envoyant une commande à un port de commande.

Pour plus d'informations sur l'administration de MQIPT à l'aide de la commande **mqiPTAdmin**, voir [«administration de MQIPT via la ligne de commande»](#), à la page 587.

## Spécification de la clé de chiffrement de mot de passe

Si la configuration MQIPT contient des mots de passe chiffrés à l'aide d'une clé de chiffrement autre que la clé par défaut, vous devez fournir la clé de chiffrement de mot de passe dans un fichier que MQIPT peut lire lorsqu'il démarre.

### Fichier de clés de chiffrement de mot de passe

Les mots de passe qui sont chiffrés pour être stockés et utilisés par MQIPT peuvent être chiffrés à l'aide d'une clé de chiffrement que vous fournissez. Si vous ne fournissez pas de clé de chiffrement, la clé de chiffrement par défaut est utilisée. Vous n'avez pas besoin de spécifier une clé de chiffrement de mot de passe, mais cette opération est plus sécurisée. Si vous ne spécifiez pas votre propre clé de chiffrement, la clé de chiffrement par défaut est utilisée.

Si vous fournissez une clé de chiffrement de mot de passe, elle doit être stockée dans un fichier accessible par la commande **mqiPTPW** utilisée pour chiffrer les mots de passe et MQIPT. Les seules restrictions concernant le contenu du fichier sont qu'il doit contenir au moins un caractère et une seule ligne de texte.

**Remarque :** Vous devez vous assurer que les droits d'accès aux fichiers appropriés sont définis sur le fichier de clé de chiffrement de mot de passe pour empêcher les utilisateurs non autorisés de lire la clé de chiffrement. Seul l'utilisateur qui exécute la commande **mqiPTPW** et l'utilisateur sous lequel MQIPT s'exécute doivent être autorisés à lire la clé de chiffrement de mot de passe.

La même clé de chiffrement de mot de passe est utilisée pour chiffrer et déchiffrer tous les mots de passe stockés pour une instance de MQIPT. Par conséquent, vous n'avez besoin que d'un seul fichier de clé de chiffrement de mot de passe pour chaque installation MQIPT.

Si la clé de chiffrement de mot de passe d'une installation MQIPT est modifiée, tous les mots de passe chiffrés doivent être rechiffrés à l'aide de la nouvelle clé de chiffrement.

## Démarrage MQIPT

Le nom par défaut du fichier de clés de chiffrement de mot de passe est `MQIPT_HOME_DIR/mqipt_cred.key`, où `MQIPT_HOME_DIR` est le répertoire dans lequel le fichier de configuration `mqipt.conf` est stocké. Si vous prévoyez d'exécuter MQIPT en tant que service démarré automatiquement, vous devez créer le fichier de clé de chiffrement de mot de passe avec le nom par défaut.

Si le fichier de clé de chiffrement de mot de passe est créé avec un nom autre que le nom par défaut, le nom du fichier doit être fourni à MQIPT lorsqu'il est démarré. Le nom du fichier de clés de chiffrement de mot de passe peut être spécifié à l'aide de l'une des méthodes suivantes, par ordre de préférence:

1. le paramètre **-sf** de la commande **mqipt** utilisée pour démarrer MQIPT.
2. la variable d'environnement `MQS_MQIPTCRED_KEYFILE`.
3. la propriété `com.ibm.mq.ipc.cred.keyfile` Java.

Si aucun nom de fichier de clé de chiffrement de mot de passe n'est fourni, le nom de fichier par défaut est utilisé, si le fichier existe. Si le fichier de clé de chiffrement de mot de passe par défaut n'existe pas, la clé de chiffrement de mot de passe par défaut est utilisée.

## administration de MQIPT via la ligne de commande

Vous pouvez utiliser la commande **mqiptAdmin** sur la ligne de commande pour administrer MQIPT.

Vous pouvez utiliser la commande **mqiptAdmin** pour exécuter les fonctions d'administration suivantes:

- Répertorier les instances locales actives de MQIPT.
- Actualisez une instance de MQIPT après avoir apporté des modifications au fichier de configuration.
- Arrêtez une instance de MQIPT.

La commande **mqiptAdmin** se trouve dans le sous-répertoire `bin` du répertoire d'installation MQIPT.

La commande **mqiptAdmin** se connecte à l'instance active de MQIPT pour l'administrer à l'aide de l'une des méthodes suivantes:

- en effectuant une connexion réseau à un port de commande.
- en se connectant à une instance locale de MQIPT sans utiliser le port de commande.

La commande **mqiptAdmin** est compatible avec les versions précédentes de MQIPT, mais vous ne pouvez pas l'utiliser pour administrer des versions de MQIPT qui sont une version supérieure à la version de la commande **mqiptAdmin**. Dans un environnement qui inclut différentes versions de MQIPT, vous devez utiliser la dernière version de la commande **mqiptAdmin**.

Pour plus d'informations sur la syntaxe de la commande **mqiptAdmin**, voir [mqiptAdmin \(administration de MQIPT\)](#).

## Administration locale sans port de commande

Les instances locales de MQIPT peuvent être administrées sans utiliser de port de commande. L'administration locale vous permet d'administrer MQIPT à l'aide de la commande **mqiptAdmin** uniquement lorsqu'elle est exécutée sur le même système que l'instance MQIPT que vous souhaitez administrer.

Pour que **mqiptAdmin** soit autorisé à administrer une instance locale de MQIPT sans utiliser le port de commande, l'instance MQIPT doit s'exécuter sur le même système et sous le même ID utilisateur que **mqiptAdmin**. Sous AIX and Linux, **mqiptAdmin** peut également être exécuté en tant que `root`.

L'administration locale est activée par défaut. Pour désactiver l'administration locale, utilisez la propriété de configuration **LocalAdmin**. Pour plus d'informations sur la propriété **LocalAdmin**, voir [LocalAdmin](#).

Pour administrer les instances locales de MQIPT, vous devez attribuer un nom à chaque instance. Vous pouvez affecter un nom à une instance de MQIPT à l'aide du paramètre **-n** lorsque vous démarrez MQIPT avec la commande **mqipt**. Si vous ne spécifiez pas de nom lors du démarrage de MQIPT, le nom du

répertoire de base est utilisé comme nom de l'instance MQIPT . Par exemple, la commande suivante démarre MQIPT et affecte le nom `ipt1` à l'instance:

```
mcipt /opt/mcipt1 -n ipt1
```

Une fois que l'instance a un nom, vous pouvez l'administrer en spécifiant le nom dans la commande **mciptAdmin** avec le paramètre **-n** . Par exemple, la commande suivante arrête l'instance locale de MQIPT avec le nom `ipt1`:

```
mciptAdmin -stop -n ipt1
```

Vous pouvez répertorier toutes les instances actives locales de MQIPT que la commande **mciptAdmin** est autorisée à administrer sans utiliser de port de commande à l'aide de la commande **mciptAdmin** avec le paramètre **-list** . Par exemple, la commande suivante répertorie toutes les instances actives locales de MQIPT que l'utilisateur qui a lancé la commande **mciptAdmin** est autorisé à administrer:

```
mciptAdmin -list
```

## Administration à l'aide d'un port de commande

Vous pouvez configurer MQIPT avec un port de commande non sécurisé et un port de commande sécurisé avec TLS. Vous pouvez utiliser ces ports de commande pour administrer MQIPT en tant qu'utilisateur se trouvant sur le même système que l'instance MQIPT que vous souhaitez administrer ou à partir d'un système distant.

Les versions précédentes de MQIPT n'acceptaient que les commandes d'administration émises vers le port de commande non sécurisé.

**Remarque :** Les connexions au port de commande non sécurisé ne sont pas chiffrées. Par conséquent, les données envoyées sur le réseau au port de commande non sécurisé, y compris le mot de passe d'accès MQIPT , peuvent être visibles pour les autres utilisateurs du réseau.

Pour que MQIPT écoute sur un port de commande les commandes émises par la commande **mciptAdmin** , une valeur doit être spécifiée pour les propriétés **CommandPort** ou **SSLCommandPort** dans la section globale du fichier de configuration `mcipt.conf` .

Passez en revue les considérations de sécurité dans [Autres considérations de sécurité](#) avant d'activer l'un des ports de commande MQIPT . Envisagez d'activer l'authentification pour les commandes reçues par les ports de commande. Pour plus d'informations sur l'authentification par port de commande, voir «Authentification du port de commande», à la page 592.

Pour administrer une instance de MQIPT à l'aide d'un port de commande, indiquez l'adresse réseau de l'hôte sur lequel MQIPT s'exécute, ainsi que le numéro de port de commande, en tant que paramètres de la commande **mciptAdmin** . Par exemple, pour actualiser l'instance MQIPT qui s'exécute sous `mcipt.example.comet` dont le port de commande non sécurisé est configuré pour être à l'écoute sur le port 1890, exécutez la commande suivante:

```
mciptAdmin -refresh -r mcipt.example.com:1890
```

Si vous ne spécifiez pas le nom d'hôte et le numéro de port, **mciptAdmin** tente de se connecter à `localhost`, port 1881.

Pour plus d'informations sur l'administration de MQIPT à l'aide du port de commande TLS, voir «Administration d' MQIPT à l'aide du port de commande TLS», à la page 588.

## Administration d' MQIPT à l'aide du port de commande TLS

MQIPT peut être configuré pour utiliser un port de commande TLS afin d'écouter les commandes d'administration émises par la commande **mciptAdmin** . L'utilisation du port de commande TLS protège les données sensibles telles que le mot de passe d'accès MQIPT sur le réseau entre **mciptAdmin** et MQIPT. Utilisez cette procédure pour configurer le port de commande TLS et administrer MQIPT à l'aide du port de commande TLS.

## Pourquoi et quand exécuter cette tâche

Le port de commande TLS doit être configuré avec un certificat serveur stocké dans un magasin de clés PKCS #12 ou dans du matériel cryptographique prenant en charge l'interface de jeton cryptographique PKCS #11 . Le certificat du serveur de port de commande est envoyé à la commande **mqiptAdmin** lors de l'établissement de liaison TLS. Cette tâche suppose que vous demandez un nouveau certificat serveur à une autorité de certification (CA) de confiance et que le certificat vous est renvoyé dans un fichier. La commande **mqiptAdmin** valide le certificat de port de commande à l'aide du certificat de l'autorité de certification qui a signé le certificat du serveur. Le certificat de l'autorité de certification doit être stocké dans un magasin de clés PKCS #12 accessible à l'aide de la commande **mqiptAdmin** .

L'authentification par certificat client n'est pas prise en charge par le port de commande TLS. Pour activer l'authentification pour les commandes d'administration émises sur un port de commande, voir «Authentification du port de commande», à la page 592.

Cette procédure explique comment gérer les magasins de clés et les certificats numériques requis pour utiliser le port de commande TLS à l'aide de la commande **V 9.4.0 V 9.4.0 mqiptKeytool** . Pour plus d'informations sur la gestion des magasins de clés utilisés par MQIPT , voir [Gestion des magasins de clés MQIPT](#).

## Procédure

1. Procédez comme suit pour configurer le port de commande TLS pour l'instance de MQIPT.

- a) Créez une paire de clés publique et privée et un certificat de serveur de port de commande TLS associé dans un magasin de clés PKCS #12 .

**V 9.4.0 V 9.4.0** Pour créer le magasin de clés qui contient le certificat du serveur de port de commande TLS, entrez la commande suivante:

```
mqiptKeytool -genkeypair -keystore filename -storetype pkcs12 -storepass password
             -dname distinguished_name -alias label
             -keyalg key_algorithm -keysize key_size -sigalg sig_algorithm
```

où :

**-keystore nom\_fichier**

Indique le nom du magasin de clés.

**-storepass mot\_de\_passe**

Indique le mot de passe du magasin de clés.

**-alias libellé**

Indique le libellé du certificat.

**-keyalg algorithme\_clé**

Indique l'algorithme utilisé pour créer la paire de clés.

**-keysize taille\_clé**

Indique la taille de la clé.

**-sigalg algorithme**

Indique l'algorithme utilisé pour signer le certificat.

**-dname nom\_distinctif**

Indique le nom distinctif X.500 entre guillemets.

- b) Créez une demande de certificat pour le certificat de serveur de port de commande TLS signé par l'autorité de certification.

**V 9.4.0 V 9.4.0** Pour créer une demande de certificat, entrez la commande suivante:

```
mqiptKeytool -certreq -keystore filename -storetype pkcs12 -storepass password
             -alias label -file certreq_filename
```

où :

**-keystore *nom\_fichier***

Indique le nom du magasin de clés.

**-storepass *mot\_de\_passe***

Indique le mot de passe du magasin de clés.

**-alias *libellé***

Indique le libellé du certificat.

**-file *nom\_fichier\_certreq***

Indique le nom de fichier de la demande de certificat.

- c) Envoyez le fichier de demande de certificat créé à l'étape «1.b», à la page 589 à votre autorité de certification à signer.
- d) Une fois que l'autorité de certification vous a envoyé le certificat signé, recevez le certificat signé dans le magasin de clés.

 Pour recevoir le certificat signé dans le magasin de clés, entrez la commande suivante:

```
mqiptykeytool -importcert -keystore cert_filename -storetype pkcs12 -storepass password  
-file cert_filename
```

où *cert\_filename* est le nom du fichier qui contient le certificat, *filename* est le nom du magasin de clés et *password* est le mot de passe du magasin de clés.

- e) Chiffrez le mot de passe du magasin de clés à l'aide de la commande **mqiptyPW**.

Entrez la commande suivante :

```
mqiptyPW -sf encryption_key_file
```

où *encryption\_key\_file* est le nom d'un fichier qui contient la clé de chiffrement de mot de passe pour votre installation MQIPT. Vous n'avez pas besoin de spécifier le paramètre **-sf** si votre installation MQIPT utilise la clé de chiffrement de mot de passe par défaut. Entrez le mot de passe du magasin de clés à chiffrer lorsque vous y êtes invité.


Pour plus d'informations sur la commande **mqiptyPW**, voir [Chiffrement d'un mot de passe de fichier de clés](#).

- f) Editez le fichier de configuration `mqipty.conf` et spécifiez les propriétés suivantes pour configurer le port de commande TLS:
- Définissez la valeur de la propriété **SSLCommandPort** sur le numéro de port de la commande TLS.
  - Définissez la valeur de la propriété **SSLCommandPortKeyRing** sur le nom de fichier du magasin de clés créé à l'étape «1.a», à la page 589.
  - Définissez la valeur de **SSLCommandPortKeyRingPW** sur la chaîne générée par la commande **mqiptyPW** à l'étape «1.e», à la page 590.
  - Définissez la valeur de la propriété **SSLCommandPortSiteLabel1** sur le nom de libellé du certificat de port de commande TLS, spécifié lors de la création de la demande de certificat à l'étape «1.b», à la page 589.
  - Si vous souhaitez restreindre les connexions entrantes au port de commande TLS à celles provenant d'une interface réseau particulière, définissez la valeur de la propriété **SSLCommandPortListenerAddress** sur une adresse réseau appartenant à l'une des interfaces réseau du système sur lequel MQIPT s'exécute. Par exemple, pour limiter les connexions entrantes au port de commande TLS à celles provenant uniquement de la machine locale, définissez la valeur de la propriété **SSLCommandPortListenerAddress** sur `localhost`.
- g) Démarrez ou actualisez MQIPT pour activer le port de commande TLS.

MQIPT émet des messages de console tels que les suivants pour afficher la configuration de port de commande TLS en cours:

```
MQCPI155 Listening for control commands on port 1882 on local address * using TLS
MQCPI139 .....secure socket protocols <NULL>
MQCPI031 .....cipher suites <NULL>
MQCPI032 .....key ring file c:\\iptHome\\ssl\\commandport.p12
MQCPI072 .....and certificate label mqiptadmin
```

2. Sur le système où la commande **mqiptAdmin** est utilisée pour administrer MQIPT, procédez comme suit pour permettre à **mqiptAdmin** de se connecter au port de commande TLS.
  - a) Importez le certificat de l'autorité de certification qui a signé le certificat de port de commande TLS dans un magasin de clés PKCS #12 à utiliser comme magasin de clés de confiance par la commande **mqiptAdmin**.

 Pour importer le certificat de l'autorité de certification, entrez la commande suivante:

```
mqiptKeytool -importcert -keystore filename -storetype pkcs12 -storepass password
             -file cert_filename -alias certlabel
```

où :

**nom de fichier**

Indique le nom du magasin de clés à créer

**Mot de passe**

Indique le mot de passe du magasin de clés

**certlabel**

Indique le libellé à attribuer au certificat de l'autorité de certification

**nom\_fichier\_certificat**

Indique le nom du fichier contenant le certificat de l'autorité de certification

- b) Chiffrez le mot de passe du magasin de clés à l'aide de la commande **mqiptPW**.

Entrez la commande suivante :

```
mqiptPW -sf encryption_key_file
```

où *encryption\_key\_file* est le nom du fichier qui contient la clé de chiffrement de mot de passe. Le fichier de clés de chiffrement de mot de passe peut être différent de celui utilisé pour chiffrer les mots de passe dans la configuration MQIPT. La clé de chiffrement de mot de passe par défaut est utilisée si vous ne spécifiez pas de fichier de clé de chiffrement avec le paramètre **-sf**. Entrez le mot de passe du magasin de clés à chiffrer lorsque vous y êtes invité.

Pour plus d'informations sur la commande **mqiptPW**, voir [Chiffrement d'un mot de passe de fichier de clés](#).

- c) Créez un fichier de propriétés à utiliser par la commande **mqiptAdmin** et spécifiez les propriétés suivantes:

```
SSLClientCAKeyRing=key_ring_file_name
SSLClientCAKeyRingPW=key_ring_password
PasswordProtectionKeyFile=encryption_key_file
```

où :

**nom\_fichier\_key\_ring\_nom\_fichier**

est le nom du magasin de clés créé à l'étape «2.a», à la page 591.

**mot\_de\_passe\_fichier\_clés**

est la sortie du mot de passe chiffré générée par la commande **mqiptPW** à l'étape «2.b», à la page 591.

## ENCRYPTION\_KEY\_FILE

est le nom du fichier qui contient la clé de chiffrement de mot de passe. Vous devez spécifier la propriété **PasswordProtectionKeyFile** uniquement si un fichier de clé de chiffrement a été utilisé pour chiffrer le mot de passe du magasin de clés à l'étape «2.b», à la page 591.

- d) Exécutez la commande **mqiptAdmin** pour administrer MQIPT en spécifiant le paramètre **-s** pour indiquer qu'une connexion TLS est requise et le paramètre **-p** pour spécifier le nom du fichier de propriétés créé à l'étape «2.c», à la page 591.

Par exemple, entrez la commande suivante pour actualiser une instance de MQIPT en envoyant une commande d'actualisation au port de commande TLS:

```
mqiptAdmin -refresh -r hostname:port -s -p properties_file
```

La commande **mqiptAdmin** émet un message du type suivant pour confirmer que la connexion à MQIPT est protégée par TLS:

```
MQCAI109 The connection to MQIPT is secured with TLSv1.2.
```

## Que faire ensuite

Pour activer l'authentification pour les commandes reçues par le port de commande TLS, suivez les étapes de la rubrique «Authentification du port de commande», à la page 592.

## Authentification du port de commande

MQIPT peut être configuré pour authentifier les commandes reçues par le port de commande non sécurisé et le port de commande TLS à l'aide d'un mot de passe. Utilisez cette procédure pour activer l'authentification du port de commande.

## Pourquoi et quand exécuter cette tâche

La commande **mqiptAdmin** invite les utilisateurs à entrer un mot de passe lorsque la commande se connecte au port de commande d'une instance de MQIPT pour laquelle l'authentification du port de commande est activée. MQIPT valide le mot de passe entré dans la commande **mqiptAdmin** par rapport au mot de passe d'accès spécifié dans la configuration MQIPT .

Les propriétés que vous définissez pour l'authentification du port de commande s'appliquent à la fois au port de commande TLS et au port de commande non sécurisé.

## Procédure

1. Chiffrez le mot de passe d'accès MQIPT à l'aide de la commande **mqiptPW** .

Entrez la commande suivante :

```
mqiptPW -sf encryption_key_file
```

où *encryption\_key\_file* est le nom du fichier qui contient la clé de chiffrement de mot de passe pour votre installation MQIPT . Vous n'avez pas besoin de spécifier le paramètre **-sf** si votre installation MQIPT utilise la clé de chiffrement de mot de passe par défaut. Entrez le mot de passe d'accès à chiffrer lorsque vous y êtes invité.

Pour plus d'informations sur le chiffrement des mots de passe dans la configuration MQIPT , voir [Chiffrement des mots de passe stockés](#).

2. Editez le fichier de configuration `mqipt.conf` et spécifiez les propriétés suivantes:

```
AccessPW=encrypted_password  
RemoteCommandAuthentication=auth_setting
```

où :



**mot\_de\_passe\_chiffré**

est la sortie du mot de passe chiffré générée par la commande **mqiPTPW** à l'étape «1», à la page 592.

**paramètre\_auth**

est l'exigence d'authentification. L'authentification du port de commande est activée si cette propriété est définie sur l'une des valeurs suivantes:

**facultatif**

Un mot de passe n'est pas requis, mais s'il est fourni, il doit être valide. Cette option peut être utile lors de la migration, par exemple.

**obligatoire**

Un mot de passe valide doit être fourni avec chaque commande reçue par un port de commande.

Pour plus d'informations sur ces propriétés, voir [Propriétés globalesMQIPT](#).

**3. Démarrez ou actualisez MQIPT pour appliquer les modifications.**

MQIPT émet un message indiquant si l'authentification du port de commande est activée. Par exemple, si MQIPT est configuré pour exiger la saisie d'un mot de passe valide chaque fois que la commande **mqiPTAdmin** est exécutée, le message suivant est émis:

```
MQCPI021 Password checking has been enabled on the command port
```

## Exécution des sauvegardes

Vous devez sauvegarder plusieurs fichiers MQIPT dans le cadre de vos procédures de sauvegarde régulières.

Sauvegardez les fichiers suivants sur une base régulière :

- Le fichier de configuration, `mqiPT.conf`
- Les fichiers de clés SSL/TLS spécifiés par les propriétés suivantes dans `mqiPT.conf`:
  - **SSLClientKeyRing**
  - **SSLClientCAKeyRing**
  - **SSLServerKeyRing**
  - **SSLServerCAKeyRing**
  - **SSLCommandPortKeyRing**
- Les fichiers de mots de passe de fichier de clés SSL/TLS spécifiés par les propriétés suivantes dans `mqiPT.conf`:
  - **SSLClientKeyRingPW**
  - **SSLClientCAKeyRingPW**
  - **SSLServerKeyRingPW**
  - **SSLServerCAKeyRingPW**
- Le fichier de clé de chiffrement de mot de passe, si la configuration MQIPT contient des mots de passe chiffrés avec une clé de chiffrement autre que la clé par défaut.
- Le fichier de règles spécifié par **SecurityManagerPolicy**, si cette propriété a été définie.
- Les fichiers d'exit de sécurité et les fichiers d'exit de certificat spécifiés par les propriétés suivantes dans `mqiPT.conf`:
  - **SecurityExitName**
  - **SSLExitName**
- Fichiers journaux de connexion dans le sous-répertoire `log` du répertoire de base MQIPT, s'ils sont nécessaires à des fins d'audit.

## Optimisation des performances

Vous pouvez optimiser les performances relatives de chaque route MQIPT à l'aide d'une combinaison d'un pool d'unités d'exécution et d'une spécification de délai d'inactivité.

### Unités d'exécution de connexion

Chaque route MQIPT se voit affecter un pool de travail d'unités d'exécution simultanées qui traitent les demandes de communication entrantes. Lors de l'initialisation, un pool d'unités d'exécution est créé (la taille des unités est définie dans l'attribut `MinConnectionThreads` de la route), et une unité d'exécution chargée de gérer la première demande entrante est affectée. Lorsque cette demande arrive, une autre unité d'exécution est affectée, prête à traiter la prochaine demande entrante. Lorsque toutes les unités d'exécution sont affectées, une nouvelle unité d'exécution est créée, ajoutée au pool de travail et affectée.

De cette manière, le pool s'agrandit jusqu'à ce que le nombre maximal d'unités d'exécution (défini dans **MaxConnectionThreads**) soit atteint. Les unités d'exécution sont retournées au pool lorsqu'une conversation prend fin, ou lorsque le délai d'inactivité spécifié s'est écoulé. Lorsque le nombre maximal d'unités d'exécution de travail est atteint, la prochaine demande entrante attend jusqu'à ce qu'une unité d'exécution soit retournée au pool de travail.

Vous pouvez réduire le temps d'attente des demandes en augmentant le nombre d'unités d'exécution disponibles. Toutefois, vous devez équilibrer cette augmentation avec les ressources système disponibles.

### Délai d'inactivité

Par défaut, les unités d'exécution ne sont pas arrêtées en raison de l'inactivité. Lorsqu'une unité d'exécution a été affectée à une conversation, elle le reste jusqu'à ce que celle-ci est fermée normalement, ou que la route soit désactivée, ou que MQIPT soit arrêté. (Facultatif) Vous pouvez spécifier un intervalle de délai d'attente d'inactivité (en minutes) dans la propriété **IdleTimeout**. De cette façon, les unités d'exécution inactives pendant la période spécifiée sont recyclées. Les unités d'exécution sont recyclées en vue d'être réutilisées sont replacées dans le pool de travail.

Si l'activité d'IBM MQ est intermittente, définissez son intervalle de pulsations sur une valeur inférieure à celle du délai d'attente de MQIPT, afin que les unités d'exécution ne soient pas constamment recyclées.

## Remarques

---

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Consultez votre représentant IBM local pour obtenir des informations sur les produits et services actuellement disponibles dans votre région. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout produit, programme ou service fonctionnellement équivalent qui ne porte pas atteinte à un droit de propriété intellectuelle IBM peut être utilisé à la place. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour obtenir des informations sur les licences relatives aux informations sur deux octets (DBCS), contactez le service de la propriété intellectuelle IBM de votre pays ou envoyez vos demandes de renseignements, par écrit, à :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales.** LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et/ou programmes décrits dans ce document.

Les références à des sites Web non IBM sont fournies uniquement à titre d'information et n'impliquent en aucune façon une adhésion de ces sites Web. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation  
Coordinateur d'interopérabilité logicielle, département 49XA  
3605 Autoroute 52 N

Rochester, MN 55901  
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans le présent document et tous les éléments sous disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Contrat sur les produits et services IBM, aux Conditions Internationales d'Utilisation de Logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Licence sur les droits d'auteur :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

## Documentation sur l'interface de programmation

---

Les informations d'interface de programmation, si elles sont fournies, sont destinées à vous aider à créer un logiciel d'application à utiliser avec ce programme.

Ce manuel contient des informations sur les interfaces de programmation prévues qui permettent au client d'écrire des programmes pour obtenir les services d'IBM MQ.

Toutefois, lesdites informations peuvent également contenir des données de diagnostic, de modification et d'optimisation. Ces données vous permettent de déboguer votre application.

**Important :** N'utilisez pas ces informations de diagnostic, de modification et d'optimisation en tant qu'interface de programmation car elles sont susceptibles d'être modifiées.

## Marques

---

IBM, le logo IBM, ibm.com, sont des marques d'IBM Corporation dans de nombreux pays. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark".

information"www.ibm.com/legal/copytrade.shtml. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.

Microsoft et Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Ce produit inclut des logiciels développés par le projet Eclipse (<https://www.eclipse.org/>).

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.







Référence :

(1P) P/N: