

9.2

*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 329.

Cette édition s'applique à la version 9 édition 2 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 qui lui semble appropriée, sans aucune obligation de votre part.

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

---

# Table des matières

<b>Présentation technique.....</b>	<b>5</b>
Présentation de la mise en file d'attente de messages.....	5
Fonctions principales et avantages de la mise en file d'attente de messages.....	7
Terminologie relative à la mise en file d'attente de messages.....	9
Messages et files d'attente.....	12
Objets IBM MQ.....	14
Types d'objet.....	15
Attribution de nom aux objets IBM MQ.....	37
Attributs d'objet.....	44
Groupes de partage de files d'attente.....	44
Objets par défaut du système.....	44
Mise en file d'attente répartie et clusters.....	45
Composants de la mise en file d'attente répartie.....	49
Composants de cluster.....	59
Messagerie de type publication/abonnement.....	66
Composants de publication/abonnement.....	66
Exemple de configuration de publication/abonnement à un seul gestionnaire de files d'attente....	94
Réseaux de publication/abonnement répartis.....	95
Multidiffusion IBM MQ.....	114
Concepts multidiffusion initiaux.....	114
MQ Telemetry - Généralités.....	116
Présentation de MQ Telemetry.....	117
Scénarios d'utilisation de la télémétrie.....	119
Connexion de dispositifs de télémétrie à un gestionnaire de files d'attente.....	126
Protocoles de connexion pour la télémétrie.....	126
Service de télémétrie (MQXR).....	126
Canaux de télémétrie.....	127
Protocole IBM MQ Telemetry Transport.....	127
Clients MQTT.....	128
Envoi d'un message à un client MQTT.....	128
Envoi d'un message à une application IBM MQ depuis un client MQTT.....	138
Applications de publication/abonnement MQTT.....	139
Applications de télémétrie.....	139
Intégration de MQ Telemetry aux gestionnaires de files d'attente.....	140
Sessions avec état et sans état MQTT.....	143
Lorsqu'un client MQTT n'est pas connecté.....	143
Couplage non contraignant entre les clients MQTT et les applications IBM MQn.....	143
Sécurité MQ Telemetry.....	144
Globalisation MQ Telemetry.....	145
Performance et évolutivité de MQ Telemetry.....	145
Périphériques compatibles avec MQ Telemetry.....	148
Sécurité dans IBM MQ.....	148
Prise en charge de TLS par le client géré IBM MQ.NET.....	149
Clients et serveurs.....	150
Présentation des IBM MQ MQI clients.....	151
Gestion et prise en charge des transactions.....	158
Extension des fonctions du gestionnaire de files d'attente.....	159
Interfaces de langage IBM MQ Java.....	160
IBM MQ classes for JMS.....	161
Fournisseur de messagerie IBM MQ.....	169
Concepts d'IBM MQ for z/OS.....	169
Le gestionnaire de files d'attente sur z/OS.....	170

L'initiateur de canal sous z/OS.....	171
Termes et tâches pour la gestion d'IBM MQ for z/OS.....	173
Files d'attente partagées et groupes de partage de files d'attente.....	176
Mise en file d'attente intragroupe.....	227
Gestion du stockage sur z/OS.....	242
Journalisation dans IBM MQ for z/OS.....	246
Définition de votre système sur z/OS.....	259
Reprise et redémarrage sur z/OS.....	270
Concepts de sécurité dans IBM MQ for z/OS.....	289
Disponibilité sur z/OS.....	296
Surveillance et statistiques sous IBM MQ for z/OS.....	300
Disposition d'une unité de récupération sur z/OS.....	301
IBM MQ et autres produits z/OS.....	304
IBM MQ et CICS.....	304
IBM MQ et IMS.....	306
IBM MQ et les adaptateurs z/OS Batch, TSO et RRS.....	310
IBM MQ for z/OS et WebSphere Application Server.....	311
Managed File Transfer.....	312
Fonctionnement de MFT avec IBM MQ.....	314
Présentation de la topologie MFT.....	315
Présentation de MFT REST API.....	316
IBM MQ Internet Pass-Thru.....	316
Utilisations de MQIPT.....	317
Fonctionnement de MQIPT.....	320
Configurations possibles de MQIPT.....	321
Configurations compatibles.....	323
Configurations de canal prises en charge.....	325
Arrêt d'un canal et conditions d'échec.....	326
Sécurité des messages.....	326
Gestionnaires de files d'attente multi-instances et haute disponibilité.....	326
<b>Remarques.....</b>	<b>329</b>
Documentation sur l'interface de programmation.....	330
Marques.....	330

# Présentation technique du IBM MQ

---

IBM MQ permet de connecter vos applications et de gérer la distribution des informations au sein de votre organisation.

IBM MQ permet aux programmes de communiquer entre eux sur un réseau de composants différents (processeurs, systèmes d'exploitation, sous-systèmes et protocoles de communication) à l'aide d'une même interface de programmation cohérente. Les applications conçues via cette interface sont connues sous le nom d'applications de mise en file d'attente de messages.

Utilisez les sous-rubriques suivantes pour en savoir plus sur la mise en file d'attente des messages et les autres fonctions d'IBM MQ.

## **Concepts associés**

[Présentation de IBM MQ](#)

## **Tâches associées**

[Planification d'une architecture IBM MQ](#)

## **Référence associée**

«Fonctions principales et avantages de la mise en file d'attente de messages», à la page 7

Ces informations mettent en évidence quelques fonctions et avantages de la mise en file d'attente des messages. Elles décrivent les fonctions telles que la sécurité et l'intégrité des données de la mise en file d'attente de messages.

[Où trouver des informations sur les exigences liées au produit et sur le support ?](#)

## Présentation de la mise en file d'attente de messages

---

Les produits IBM MQ permettent aux programmes de communiquer entre eux sur un réseau de composants disparates (processeurs, systèmes d'exploitation, sous-systèmes et protocoles de communication) en utilisant une interface de programmation d'application cohérente.

Les applications conçues et écrites à l'aide de cette interface sont appelées applications de *mise en file d'attente de messages*, car elles utilisent le style *messagerie* et *mise en file d'attente* :

- Messagerie signifie que les programmes communiquent en s'envoyant des données dans des messages plutôt qu'en s'appelant directement.
- Mise en file d'attente signifie que les messages sont stockés dans des files d'attente, pour que les programmes puissent être exécutés indépendamment les uns des autres, à des vitesses et des heures différentes et dans des emplacements distincts, sans être connectés de manière logique.

La mise en file d'attente de messages a été utilisée dans le traitement de l'information pendant de nombreuses années. Elle est plus couramment utilisée aujourd'hui dans la messagerie électronique. Sans le système de mise en file d'attente, l'envoi d'un message électronique sur de longues distances nécessite que chaque noeud du chemin soit disponible pour l'acheminement des messages et que les adresses soient connectées et informées du fait que vous tentez de leur envoyer un message. Dans un système de mise en file d'attente, les messages sont stockés sur des noeuds intermédiaires jusqu'à ce que le système soit prêt à les acheminer. A leur destination finale, ils sont stockés dans une boîte aux lettres électronique jusqu'à ce que le destinataire soit prêt à les lire.

Toute fois, la plupart des transactions commerciales complexes sont traitées aujourd'hui sans le système de mise en file d'attente. Dans un réseau de grande taille, il se peut que le système gère plusieurs milliers de connexions prêtes à être utilisées. Si une partie du système rencontre un problème, de nombreuses parties du système deviennent inutilisables.

Vous pouvez considérer la mise en file d'attente de messages comme étant une messagerie électronique pour les programmes. Dans un environnement de mise en file d'attente de messages, chaque programme faisant partie d'une suite d'applications exécute une fonction autonome bien définie en réponse à une demande spécifique. Pour communiquer avec un autre programme, un programme doit placer un

message dans une file d'attente prédéfinie. L'autre programme extrait le message de la file d'attente et traite les demandes et les informations contenues dans le message. La mise en file d'attente de messages est donc un style de communication de programme à programme.

La mise en file d'attente est un mécanisme permettant de conserver les messages tant qu'une application n'est pas prête à les traiter. La mise en file d'attente vous permet :

- de communiquer entre les programmes (chacun pouvant être exécuté dans des environnements différents) sans avoir à écrire le code de communication ;
- de sélectionner l'ordre dans lequel un programme traite les messages ;
- d'équilibrer les charges sur un système en prévoyant plusieurs programmes pour la prise en charge d'une file d'attente lorsque le nombre de messages dépasse un seuil ;
- d'augmenter la disponibilité de vos applications en prévoyant un système secondaire pour la prise en charge des files d'attente si votre système principal est indisponible.

## **Description d'une file d'attente de messages**

Une file d'attente de messages, connue tout simplement sous le nom de file d'attente, est une destination nommée à laquelle les messages peuvent être envoyés. Les messages s'accumulent dans les files d'attente jusqu'à ce qu'ils soient extraits par des programmes prenant en charge ces files d'attente.

Les files d'attente résident dans un gestionnaire de files d'attente et sont gérées par celui-ci, (voir [«Terminologie relative à la mise en file d'attente de messages»](#), à la page 9). La nature physique d'une file d'attente dépend du système d'exploitation sur lequel le gestionnaire de files d'attente est en cours d'exécution. Une file d'attente peut être une zone de mémoire tampon volatile dans la mémoire d'un ordinateur ou un jeu de données sur une unité de stockage permanent (telle qu'un disque). La gestion physique des files d'attente incombe au gestionnaire de files d'attente et n'est pas apparente pour les programmes d'application participants.

Les programmes accèdent aux files d'attente uniquement via les services externes du gestionnaire de files d'attente. Ils peuvent ouvrir une file d'attente, y placer des messages, en extraire des messages et la fermer. Ils peuvent également définir et consulter les attributs des files d'attente.

## **Différents styles de mise en file d'attente de messages**

### **Point-à-point**

Un message est placé dans la file d'attente et une application reçoit ce message.

Dans la messagerie point-à-point, l'application d'envoi doit connaître certaines informations sur l'application de réception avant de pouvoir lui envoyer un message. Par exemple, elle doit connaître le nom de la file d'attente à laquelle envoyer les informations et peut également indiquer un nom de gestionnaire de files d'attente.

### **Publication/Abonnement**

Une copie de chaque message publié par une application de publication est livrée à toutes les applications intéressées. Il se peut qu'il y ait plusieurs, une ou aucune application intéressée. Dans la messagerie de publication/abonnement, une application intéressée est appelée un abonné et les messages sont placés dans une file d'attente identifiée par un abonnement.

La messagerie de type publication/abonnement vous permet de dissocier le fournisseur d'informations des consommateurs de ces informations. Il n'est pas nécessaire que l'application émettrice et l'application de réception se connaissent pour que les informations puissent être envoyées et reçues. Pour plus d'informations, voir [«Messagerie de type publication/abonnement»](#), à la page 66.

## Avantages de la mise en file d'attente de messages pour le concepteur et le développeur d'applications

IBM MQ permet aux programmes d'application d'utiliser la *mise en file d'attente des messages* pour participer à un traitement géré par message. Les programmes d'application peuvent communiquer sur des plateformes différentes à l'aide des produits logiciels appropriés de mise en file d'attente de messages. Par exemple, les applications z/OS peuvent communiquer via IBM MQ for z/OS. Les applications sont protégées du mécanisme des communications sous-jacentes. Voici quelques autres avantages de la mise en file d'attente de messages :

- Vous pouvez concevoir des applications à l'aide de petits programmes pouvant être partagés entre plusieurs applications.
- Vous pouvez générer rapidement de nouvelles applications en réutilisant ces blocs fonctionnels.
- Les applications écrites pour l'utilisation des techniques de mise en file d'attente de messages ne sont pas affectées par les modifications apportées au mode de fonctionnement des gestionnaires de files d'attente.
- Il n'est pas nécessaire d'utiliser des protocoles de communication. Le gestionnaire de files d'attente traite tous les aspects de la communication pour vous.
- Il n'est pas nécessaire que les programmes recevant des messages soient en cours d'exécution lorsque des messages leur sont envoyés. Les messages sont conservés dans des files d'attente.

Les concepteurs peuvent réduire le coût de leurs applications car le développement est plus rapide, un plus petit nombre de développeurs est requis et les demandes en compétences de programmation sont inférieures à celles des applications qui n'utilisent pas la mise en file d'attente de messages.

IBM MQ implémente une interface de programmation d'application commune appelée *message queue interface* (ou MQI), quel que soit l'emplacement d'exécution des applications. Cela facilite la portabilité des programmes d'application d'une plateforme à l'autre.

Pour plus de détails sur l'interface MQI, voir [Présentation de l'interface MQI \(Message Queue Interface\)](#).

## Fonctions principales et avantages de la mise en file d'attente de messages

Ces informations mettent en évidence quelques fonctions et avantages de la mise en file d'attente des messages. Elles décrivent les fonctions telles que la sécurité et l'intégrité des données de la mise en file d'attente de messages.

Les principales fonctions des applications qui utilisent les techniques de mise en file d'attente de messages sont les suivantes :

- Il n'existe aucune connexion directe entre les programmes.
- La communication entre les programmes peut être sans contrainte de temps.
- Le travail peut être réalisé par de petits programmes autonomes.
- La communication peut être gérée par des événements.
- Les applications peuvent attribuer une priorité à un message.
- Sécurité.
- Intégrité des données
- Prise en charge de la reprise.

### Absence de connexion directe entre les programmes

La mise en file d'attente de messages est une technique de communication indirecte de programme à programme. Elle peut être utilisée au sein de n'importe quelle application dans laquelle les programmes communiquent entre eux. La communication est établie lorsqu'un programme place des messages dans une file d'attente (détenue par un gestionnaire de files d'attente) et qu'un autre programme extrait les messages de la file d'attente.

Les programmes peuvent extraire des messages placés dans une file d'attente par d'autres programmes. Ces derniers peuvent être connectés au même gestionnaire de files d'attente que le

programme récepteur ou à un autre gestionnaire de files d'attente. Cet autre gestionnaire de files d'attente peut résider sur un autre système, un système informatique différent, voire dans un secteur d'activité ou une entreprise différente.

Il n'existe aucune connexion physique entre les programmes qui communiquent à l'aide de files d'attente de messages. Un programme envoie des messages à une file d'attente détenue par un gestionnaire de files d'attente et un autre programme extrait les messages de la file d'attente (voir Figure 1, à la page 8).

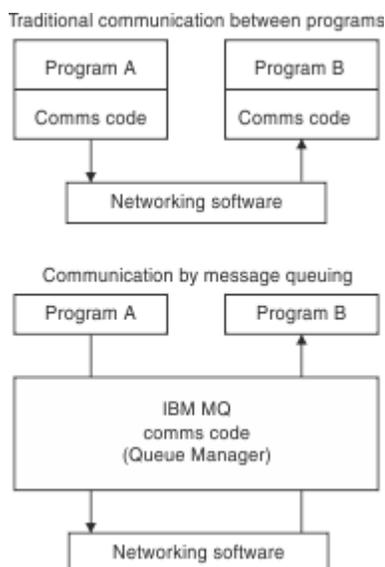


Figure 1. Comparaison de la mise en file d'attente de messages à une communication classique

Comme dans le cas de la messagerie électronique, les messages individuels faisant partie d'une transaction se déplacent via un réseau dans le cadre d'un processus de stockage et retransmission. Si un lien entre les noeuds échoue, le message est conservé jusqu'à ce que le lien soit restauré, ou bien l'opérateur ou le programme réachemine le message.

Le mécanisme permettant de déplacer un message d'une file d'attente à une autre n'est pas visible par les programmes. Par conséquent, les programmes sont plus simples.

### Communication sans contrainte de temps

Les programmes qui demandent à d'autres programmes d'effectuer un travail n'ont pas besoin d'attendre la réponse à une demande. Ils peuvent effectuer d'autres travaux et traiter la réponse lorsque celle-ci arrive ou ultérieurement. Lors de l'écriture d'une application de messagerie, vous n'avez pas besoin de connaître (ou de vous en préoccuper) à quel moment un programme envoie un message ou la cible est capable de recevoir le message. Le message n'est pas perdu ; il est conservé par le gestionnaire de files d'attente jusqu'à ce que la cible soit prête à le traiter. Le message demeure dans la file d'attente tant qu'il n'est pas supprimé par un programme. En d'autres termes, les programmes d'application émetteur et récepteur sont dissociés ; l'émetteur peut poursuivre le traitement sans attendre que le récepteur accuse réception du message. Il n'est même pas nécessaire que l'application cible soit en cours d'exécution lorsque le message est envoyé. Elle peut extraire le message une fois qu'elle a été démarrée.

### Petits programmes

La mise en file d'attente de messages vous permet de tirer parti de l'utilisation de petits programmes autonomes. A la place d'un programme unique de grande taille effectuant toutes les parties d'un travail en séquence, vous pouvez répartir le travail sur plusieurs programmes indépendants de plus petite taille. Le programme demandeur envoie des messages à chacun des programmes distincts, leur demandant d'accomplir leur fonction ; lorsque chaque programme est terminé, les résultats sont renvoyés sous la forme d'un ou de plusieurs messages.

### Traitement géré par message

A l'arrivée des messages dans une file d'attente, ceux-ci peuvent automatiquement démarrer une application à l'aide d'un *déclenchement*. Si nécessaire, les applications peuvent être arrêtées lorsque le message (ou les messages) a été traité.

### Traitement géré par événement

Les programmes peuvent être contrôlés en fonction de l'état des files d'attente. Par exemple, vous pouvez faire en sorte qu'un programme démarre dès qu'un message arrive dans une file d'attente ou vous pouvez préciser que le programme ne démarre que lorsque la file d'attente contient, par exemple, 10 messages au-dessus d'une certaine priorité ou 10 messages ayant n'importe quelle priorité.

### Priorité de message

Un programme peut attribuer une priorité à un message lorsqu'il place le message dans une file d'attente. Cela détermine l'emplacement de la file d'attente dans lequel le nouveau message est ajouté.

Les programmes peuvent extraire des messages d'une file d'attente soit dans l'ordre des messages dans la file d'attente, soit en extrayant un message spécifique. (Un programme peut être amené à extraire un message spécifique s'il recherche la réponse à une demande qu'il a envoyée plus tôt.)

### Sécurité

Des fonctions de sécurité sont fournies et incluent l'authentification des applications lorsqu'elles utilisent un gestionnaire de files d'attente, les vérifications d'autorisation lorsqu'elles utilisent des ressources comme une file d'attente sur le gestionnaire de files d'attente, et le chiffrement des données de message qui sont transmises via le réseau et qui se trouvent dans les files d'attente. Pour plus d'informations sur la sécurité, voir [Présentation de la sécurité](#).

### Intégrité des données

L'intégrité des données est assurée par des unités d'oeuvre. La synchronisation du début et de la fin des unités d'oeuvre est entièrement prise en charge sous forme d'option sur chaque MQGET ou MQPUT, ce qui permet la validation ou l'annulation des résultats de l'unité d'oeuvre. Le fonctionnement du support des points de synchronisation est interne ou externe à IBM MQ en fonction de la forme de coordination de point de synchronisation sélectionnée pour l'application.

### Prise en charge de la reprise

Pour que la reprise soit possible, toutes les mises à jour persistantes d'IBM MQ sont consignées. Si une reprise est nécessaire, tous les messages persistants sont restaurés, toutes les transactions en cours annulées et toutes les validations et annulations de point de synchronisation traitées normalement par le gestionnaire de point de synchronisation responsable. Pour plus d'informations sur les messages persistants, voir [Persistence des messages](#).

**Remarque :** En ce qui concerne les clients et les serveurs IBM MQ, il n'est pas nécessaire de modifier une application serveur pour qu'elle prenne en charge des IBM MQ MQI clients sur les supplémentaires sur les nouvelles plateformes. De même, le IBM MQ MQI client peut, sans modification, fonctionner avec d'autres types de serveurs.

## Terminologie relative à la mise en file d'attente de messages

Ces informations donnent un aperçu de certains termes utilisés dans le cadre de la mise en file d'attente de messages.

Il s'agit des termes suivants :

- [Message](#)
- [Descripteur de message](#)
- [File d'attente](#)
- [Gestionnaire de files d'attente](#)
- [Canaux](#)
- [Agent MCA \(Message Channel Agent\)](#)
- [Grappe](#)

-  [File d'attente partagée](#)
-  [Groupe de partage de files d'attente](#)
-  [Mise en file d'attente intra-groupe](#)
- [IBM MQ MQI client](#)
- [Point à point](#)
- [Publication/abonnement](#)
- [Rubrique](#)
- [Abonnement](#)

## Message

Dans le cadre de la mise en file d'attente de messages, un message est un ensemble de données envoyées par un programme et destinées à un autre programme. Voir [Messages IBM MQ](#). Pour plus d'informations sur les types de messages, voir [Types de message](#).

## Descripteur de message

Un message IBM MQ est constitué d'informations de contrôle et de données d'application.

Les informations de contrôle sont définies dans une structure de descripteur de message (MQMD) et contient les éléments suivants :

- Le type du message
- Un identificateur pour le message
- La priorité de livraison du message

La structure et le contenu des données d'application sont déterminés par les programmes participants et non par IBM MQ.

## File d'attente

Destination nommée à laquelle les messages peuvent être envoyés. Les messages s'accumulent dans les files d'attente jusqu'à ce qu'ils soient extraits par des programmes prenant en charge ces files d'attente.

## Gestionnaire de files d'attente

Un *gestionnaire de files d'attente* est un programme système qui met à la disposition des applications des services de mise en file d'attente.

Il fournit une interface de programme d'application qui permet aux programmes de placer et d'extraire des messages dans les files d'attente. Un gestionnaire de files d'attente fournit des fonctions supplémentaires permettant aux administrateurs de créer des files d'attente, de modifier les propriétés des files d'attente existantes et de contrôler le fonctionnement du gestionnaire de files d'attente.

Pour que les services de mise en file d'attente de messages IBM MQ soient disponibles sur un système, un gestionnaire de files d'attente doit être en cours d'exécution. Plusieurs gestionnaires de files d'attente peuvent être en cours d'exécution sur un seul système (par exemple, pour séparer un système de test d'un système *opérationnel*). Pour une application, chaque gestionnaire de files d'attente est identifié par un *descripteur de connexion* (*Hconn*).

Plusieurs applications différentes peuvent utiliser en même temps les services du gestionnaire de files d'attente et peuvent être entièrement non apparentées. Pour qu'un programme utilise les services d'un gestionnaire de files d'attente, il doit établir une connexion avec ce gestionnaire de files d'attente.

Pour que les applications envoient des messages à des applications connectées à d'autres gestionnaires de files d'attente, les gestionnaires de files d'attente doivent être en mesure de communiquer entre eux. IBM MQ implémente un protocole de *stockage et retransmission* garantissant la distribution sécurisée des messages entre ces applications.

## Canaux

Les canaux sont des objets qui fournissent un chemin de communication d'un gestionnaire de files d'attente vers un autre. Les canaux sont utilisés pour déplacer des messages d'un gestionnaire de files d'attente vers un autre et ils protègent les applications des protocoles de communication sous-jacents. Les gestionnaires de files d'attente peuvent résider sur la même plateforme ou sur des plateformes différentes. Les messages envoyés peuvent provenir de nombreux endroits :

- programmes d'application écrits par l'utilisateur qui transfèrent des données d'un noeud vers un autre,
- applications d'utilisateur écrites par l'utilisateur qui utilisent des commandes PCF ou WebSphere MQ Administration Interface,
- IBM MQ Explorer,
- gestionnaires de files d'attente qui envoient des messages d'événement d'instrumentation à un autre gestionnaire de files d'attente,
- gestionnaires de files d'attente qui envoient des commandes d'administration distantes à un autre gestionnaire de files d'attente. Par exemple, via l'utilisation de commandes MQSC ou de l'administrative REST API.

## Agent MCA

Un agent MCA est l'extrémité d'un canal. Une paire d'agents MCA, un émetteur et un récepteur, constituent un canal et déplacent des messages d'un gestionnaire de files d'attente à un autre.

## Cluster

Un *cluster* est un réseau de gestionnaires de files d'attente logiquement associés d'une manière ou d'une autre.

Dans un réseau IBM MQ qui utilise la mise en file d'attente répartie sans fonction de mise en cluster, chaque gestionnaire de files d'attente est indépendant. Si un gestionnaire de files d'attente doit envoyer des messages à un autre gestionnaire de files d'attente, il doit avoir défini une file d'attente de transmission et un canal vers le gestionnaire de files d'attente éloignées.

Les clusters sont utilisés pour deux raisons différentes : réduire l'administration du système et améliorer la disponibilité et l'équilibrage de charge.

Même si vous établissez un très petit cluster, vous bénéficiez d'une administration de système simplifiée. Les gestionnaires de files d'attente faisant partie d'un cluster nécessitent moins de définitions, ce qui réduit le risque d'erreur dans vos définitions.

Pour plus d'informations sur la mise en cluster, voir [Clusters](#).

### File d'attente partagée

Une *file d'attente partagée* est un type de file d'attente locale comportant des messages accessibles à un ou plusieurs gestionnaires de files d'attente d'un sysplex. Il ne s'agit pas d'une file d'attente partagée par plusieurs applications, à l'aide du même gestionnaire de files d'attente. Cette remarque est valable pour IBM MQ for z/OS uniquement.

### Groupe de partage de files d'attente

Les gestionnaires de files d'attente pouvant accéder au même ensemble de files d'attente partagées forment un groupe appelé *groupe de partage de files d'attente*. Ils communiquent entre eux à l'aide d'une unité de couplage qui stocke les files d'attente partagées. Cette remarque est valable pour IBM MQ for z/OS uniquement. Pour plus d'informations sur les groupes de partage de files d'attente, voir «Files d'attente partagées et groupes de partage de files d'attente», à la page 176.

### Mise en file d'attente intragroupe

Les gestionnaires de files d'attente d'un groupe de partage de files d'attente peuvent communiquer à l'aide de canaux normaux ou vous pouvez utiliser une technique appelée *mise en file d'attente*

*intra-groupe*, qui vous permet de transférer rapidement les messages sans définir de canaux. Cette remarque est valable pour IBM MQ for z/OS uniquement.

## IBM MQ MQI client

Les *clients* IBM MQ MQI sont des composants installables indépendamment d' IBM MQ. MQI Client vous permet d'exécuter des applications IBM MQ à l'aide d'un protocole de communication, afin d'interagir avec un ou plusieurs serveurs MQI (Message Queue Interface) sur d'autres plateformes et de se connecter à leurs gestionnaires de files d'attente.

Pour des détails complets sur l'installation et l'utilisation des composants du IBM MQ MQI client, voir les rubriques suivantes :

-  [Installation d'un client IBM MQ sous AIX](#)
-  [Installation d'un client IBM MQ sous Linux®](#)
-  [Installation d'un client IBM MQ sous Windows](#)
-  [Installation d'un client IBM MQ sous IBM i](#)

et [Configuration des connexions entre le serveur et le client.](#)

## Messagerie point-à-point

Dans la messagerie point-à-point, chaque message est envoyé depuis une application de production vers une application destinataire. Les messages sont transférés lorsque l'application de production place les messages dans une file d'attente et que l'application destinataire les extrait de cette file d'attente.

## Messagerie de type publication/abonnement

Dans une messagerie de publication/abonnement, une copie de chaque message publié par une application de publication est livrée à toutes les applications intéressées. Il se peut qu'il y ait plusieurs, une ou aucune application intéressée. Dans la messagerie de publication/abonnement, une application intéressée est appelée un abonné et les messages sont placés dans une file d'attente identifiée par un abonnement. Pour plus d'informations, voir [«Messagerie de type publication/abonnement»](#), à la page 66.

## Topic

Une rubrique est une chaîne de caractères décrivant l'objet des informations publiées dans un message de publication/abonnement.

Les rubriques sont primordiales pour la distribution des messages dans un système de publication/abonnement. Au lieu d'inclure une adresse de destination spécifique dans chaque message, un diffuseur de publications affecte une rubrique à chaque message. Le gestionnaire de files d'attente compare la rubrique à une liste d'abonnés qui s'y sont abonnés et distribue le message à chacun d'eux.

## Abonnement

Une application de publication/abonnement peut enregistrer une information présentant un intérêt particulier dans des rubriques spécifiques. Si tel est le cas, l'application est appelée un abonné et le terme abonnement décrit comment les messages concordants sont mis en file d'attente à des fins de traitement.

Un abonnement contient des informations sur l'identité de l'abonné et l'identité de la file d'attente de destination sur laquelle les publications doivent être placées. Il contient également des informations sur la méthode de placement d'une publication dans la file d'attente de destination.

## Messages et files d'attente

Les messages et les files d'attente constituent les composants de base d'un système de mise en file d'attente de messages.

## Description d'un message

Un *message* est une chaîne d'octets significative pour les applications qui l'utilisent. Les messages servent à transférer des applications d'un programme d'application à l'autre (ou entre des parties différentes de la même application). Les applications peuvent être actives sur la même plateforme ou sur des plateformes différentes.

Un message IBM MQ est constitué des éléments suivants :

- *Données d'application*. Le contenu et la structure des données d'application sont définies par les programmes d'application qui les utilisent.
- *Descripteur de message*. Le descripteur de message identifie le message et contient des informations de contrôle supplémentaires, telles que le type de message et la priorité attribuée au message par l'application émettrice.

Le format du descripteur de message est défini par IBM MQ. Pour obtenir une description complète du descripteur de message, voir [MQMD - Descripteur de message](#).

- *Les propriétés de message*. Métadonnées du message. Le contenu des propriétés de message sont définies par les programmes d'application qui les utilisent. Pour plus d'informations, voir [Propriétés de message](#).

## Longueurs de message

La longueur maximale par défaut d'un message est de 4 Mo, bien que vous puissiez l'augmenter à une longueur maximale de 100 Mo (où 1 Mo équivaut à 1 048 576 octets). En pratique, la longueur de message peut être limitée par :

- La longueur de message maximale définie pour la file d'attente réceptrice
- La longueur de message maximale définie pour le gestionnaire de files d'attente
- La longueur de message maximale définie pour la file d'attente
- La longueur de message maximale définie pour l'application émettrice ou l'application réceptrice
- L'espace de stockage disponible pour le message

Plusieurs messages peuvent être nécessaires pour l'envoi de toutes les informations requises par une application.

## Mode d'envoi et de réception des messages par les applications

Les programmes d'application envoient et reçoivent les messages à l'aide d'**appels MQI**.

Par exemple, pour placer un message dans une file d'attente, une application :

1. ouvre la file d'attente requise, en émettant un appel MQI MQOPEN ;
2. émet un appel MQI MQPUT pour placer le message dans la file d'attente.

Une autre application peut extraire le message de la même file d'attente en émettant un appel MQI MQGET.

Pour plus d'informations sur les appels MQI, voir [Appels MQI](#).

## Description d'une file d'attente

Une *file d'attente* est une structure de données destinée au stockage de messages.

Chaque file d'attente est détenue par un *gestionnaire de files d'attente*. Le gestionnaire de files d'attente est responsable de la gestion des files d'attente qui lui appartiennent et du stockage de tous les messages qu'il reçoit dans les files d'attente appropriées. Les messages peuvent être placés dans la file d'attente par des programmes d'application ou par un gestionnaire de files d'attente dans le cadre de ses opérations normales.

## Files d'attente prédéfinies et dynamiques

Les files d'attente peuvent être caractérisées par la façon dont elles sont créées :

- Les **files d'attente prédéfinies** sont créées par un administrateur à l'aide des commandes MQSC ou PCF appropriées. Les files d'attente prédéfinies sont permanentes ; elles existent indépendamment des applications qui les utilisent et sont conservées après le redémarrage d'IBM MQ.
- Les **files d'attente dynamiques** sont créées lorsqu'une application émet une demande MQOPEN indiquant le nom d'une *file d'attente modèle*. La file d'attente créée est basée sur une *définition de file d'attente modèle* appelée file d'attente modèle. Vous pouvez créer une file d'attente modèle à l'aide de la commande MQSC DEFINE QMODEL. Les attributs d'une file d'attente modèle (par exemple, le nombre maximal de messages qu'elle peut stocker) sont hérités des files d'attente dynamiques créées à partir de la file d'attente modèle.

Les files d'attente modèle comportent un attribut indiquant si la file d'attente dynamique doit être permanente ou temporaire. Les files d'attente permanentes sont conservées après le redémarrage de l'application et du gestionnaire de files d'attente ; les files d'attente temporaires sont perdues au redémarrage.

## Extraction de messages des files d'attente

Seules les applications disposant des droits d'accès appropriés peuvent extraire des messages d'une file d'attente conformément aux algorithmes d'extraction suivants :

- Premier entré, premier sorti (FIFO)
- Priorité de message, comme défini dans le descripteur de message. Les messages ayant la même priorité sont extraits suivant la méthode FIFO.
- Demande de programme pour un message spécifique.

La demande MQGET émanant de l'application détermine la méthode utilisée.

## Objets IBM MQ

---

Les gestionnaires de files d'attente définissent les propriétés des objets IBM MQ. Les valeurs de ces propriétés changent la manière dont IBM MQ traite ces objets. Vous créez et gérez des objets à l'aide de commandes et d'interfaces d'IBM MQ. Dans vos applications, vous utilisez MQI (Message Queue Interface) pour contrôler les objets. Les objets sont identifiés par un IBM MQ *descripteur d'objet* (MQOD) lorsqu'ils sont traités à partir d'un programme.

L'administration des objets impliquent les tâches suivantes :

- Démarrage et arrêt des gestionnaires de files d'attente.
- Création d'objets, en particulier des files d'attente, pour les applications.
- Affichage ou modification des attributs d'objets.
- Suppression d'objets.
- Utilisation de canaux pour créer des chemins de communication vers des gestionnaires de files d'attente sur d'autres systèmes (distants).
- Création de *clusters* de gestionnaires de files d'attente pour simplifier le processus d'administration global et pour équilibrer la charge de travail.

A l'exception des files d'attente dynamiques, les objets doivent être définis sur le gestionnaire de files d'attente avant que vous puissiez les utiliser.

Lorsque vous utilisez une commande IBM MQ pour exécuter une opération d'administration d'objet, le gestionnaire de files d'attente vérifie que vous disposez des droits nécessaires pour effectuer l'opération. De même, lorsqu'une application utilise l'appel MQOPEN pour ouvrir un objet, le gestionnaire de files d'attente vérifie que l'application dispose du niveau de droits d'accès requis avant qu'elle soit autorisée à accéder à cet objet. Les vérifications s'effectuent au niveau du nom de l'objet à ouvrir.

Vous pouvez définir et gérer des objets à l'aide des méthodes suivantes :

- Les commandes PCF décrites dans les rubriques [Référence de formats de commande programmable et Automatisation des tâches d'administration](#)
- Les commandes MQSC décrites dans [Les commandes MQSC](#)
-  Les panneaux d'opérations et de contrôle d'IBM MQ for z/OS, décrits dans [Utilisation d'IBM MQ for z/OS](#)
-   IBM MQ Explorer (Windows et Linux pour les systèmes Intel uniquement). Pour plus d'informations, voir [Présentation de MQ Explorer](#).

Vous pouvez également gérer les objets à l'aide des méthodes suivantes :

- Commandes de contrôle, saisies à partir d'un clavier. Voir [Administration à l'aide des commandes de contrôle](#).
- Appels d'interface d'administration IBM MQ (MQAI) dans un programme. Voir [IBM MQ Administration Interface \(MQAI\)](#).

 Pour les séquences de commandes IBM MQ for z/OS que vous utilisez régulièrement, vous pouvez écrire des programmes d'administration qui créent des messages contenant des commandes et qui placent ces messages dans la file d'attente d'entrée des commandes système. Le gestionnaire de files d'attente traite les messages de cette file d'attente de la même façon qu'il traite les commandes entrées à partir de la ligne de commande ou des panneaux d'opérations et de contrôle. Cette technique est décrite dans [Ecriture de programmes pour administrer IBM MQ](#), et montrée dans l'exemple d'application Mail Manager fourni avec IBM MQ for z/OS. Pour la description de cet exemple, voir [Exemples de programmes pour IBM MQ for z/OS](#).

 Pour les séquences de commandes IBM MQ pour IBM i que vous utilisez régulièrement, vous pouvez écrire des programmes CL. Pour plus d'informations, voir [Gestion d'IBM MQ for IBM i à l'aide de commandes CL](#).

 Pour les séquences de commandes IBM MQ sous AIX, Linux, and Windows, vous pouvez utiliser la fonction MQSC afin d'exécuter une série de commandes stockées dans un fichier. Pour plus d'informations, voir [Administration à l'aide des commandes MQSC](#).

## Types d'objet

La plupart des tâches d'administration impliquent la manipulation de différents types d' IBM MQ objets .

Pour plus d'informations sur l'attribution de nom aux objets IBM MQ, voir [«Attribution de nom aux objets IBM MQ»](#), à la page 37.

Pour plus d'informations sur les objets par défaut créés dans un gestionnaire de files d'attente, voir [«Objets par défaut du système»](#), à la page 44.

Pour plus d'informations sur les différents types d'objets IBM MQ, consultez les sous-rubriques suivantes :

### Concepts associés

[«Présentation de la mise en file d'attente de messages»](#), à la page 5

Les produits IBM MQ permettent aux programmes de communiquer entre eux sur un réseau de composants disparates (processeurs, systèmes d'exploitation, sous-systèmes et protocoles de communication) en utilisant une interface de programmation d'application cohérente.

[«Attributs d'objet»](#), à la page 44

Les propriétés d'un objet sont définies par ses attributs. Certains peuvent être spécifiés et d'autres ne peuvent être que consultés.

### Référence associée

[Les commandes MQSC](#)

## Files d'attente

Présentation des files d'attente et des attributs de file d'attente IBM MQ.

Une IBM MQ file d'attente est un objet nommé dans lequel les applications peuvent insérer des messages et à partir duquel les applications peuvent extraire des messages.

Les messages sont stockés dans une file d'attente de sorte que, si l'application d'insertion attend une réponse à son message, elle est disponible pour effectuer d'autres activités en attendant cette réponse. Les applications accèdent à une file d'attente à l'aide de l'interface MQI (Message Queue Interface), décrite dans la rubrique [Présentation de l'interface MQI \(Message Queue Interface\)](#).

Avant qu'un message puisse être placé dans une file d'attente, celle-ci doit avoir déjà été créée. Une file d'attente est détenue par un gestionnaire de files d'attente qui peut posséder plusieurs files d'attente. Cependant, chaque file d'attente doit porter un nom unique au sein de ce gestionnaire de files d'attente.

Une file d'attente est gérée via un gestionnaire de files d'attente. Dans la plupart des cas, chaque file d'attente est physiquement gérée par son gestionnaire de files d'attente, ce qui n'est toutefois pas apparent pour un programme d'application. Les files d'attente partagées IBM MQ for z/OS peuvent être gérées par n'importe quel gestionnaire de files d'attente dans le groupe de partage de files d'attente.

Pour créer une file d'attente, vous pouvez utiliser des commandes IBM MQ (MQSC), des commandes PCF ou des interfaces spécifiques à la plateforme. Par exemple, les opérations et les panneaux de contrôle IBM MQ for z/OS sont spécifiques à la plateforme.

Vous pouvez créer *de façon dynamique* des files d'attente pour des travaux temporaires, à partir de votre application. Par exemple, vous pouvez créer des files d'attente de *réponse* (qui ne sont plus requises une fois qu'une application prend fin). Pour plus d'informations, voir [«Files d'attente dynamiques et modèle»](#), à la page 21.

Avant d'utiliser une file d'attente, vous devez l'ouvrir, afin d'indiquer l'opération souhaitée au niveau de la file d'attente. Par exemple, vous pouvez ouvrir une file d'attente pour :

- parcourir les messages uniquement (et non les extraire) ;
- extraire des messages (et partager l'accès avec d'autres programmes ou avec un accès exclusif) ;
- insérer des messages dans la file d'attente ;
- consulter les attributs de la file d'attente ;
- définir les attributs de la file d'attente.

Pour obtenir la liste complète des options pouvant être indiquées lors de l'ouverture d'une file d'attente, voir [MQOPEN - Ouverture d'objet](#).

## Attributs des files d'attente

Certains attributs d'une file d'attente sont spécifiés lors de la définition de la file d'attente et ne peuvent pas être modifiés par la suite (par exemple, le type de la file d'attente). D'autres attributs des files d'attente peuvent être regroupés en attributs pouvant être modifiés :

- par le gestionnaire de files d'attente lors du traitement de la file d'attente (par exemple, la longueur en cours d'une file d'attente) ;
- uniquement par des commandes (par exemple, la description de la file d'attente) ;
- par des applications, à l'aide de l'appel MQSET (par exemple, si des opérations d'insertion sont autorisées sur la file d'attente).

Les valeurs de tous les attributs sont disponibles à l'aide de l'appel MQINQ.

Les attributs communs à plusieurs types de file d'attente sont les suivants :

### **nomQ**

Nom de la file d'attente.

### **QType**

Type de la file d'attente.

**QDesc**

Description de la file d'attente.

**InhibitGet**

Indique si les programmes sont autorisés à extraire des messages de la file d'attente. Toutefois, vous ne pouvez jamais extraire de messages de files d'attente éloignées.

**InhibitPut**

Indique si les programmes sont autorisés à insérer des messages dans la file d'attente.

**DefPriority**

Priorité par défaut des messages insérés dans la file d'attente.

**DefPersistence**

Persistence par défaut des messages insérés dans la file d'attente

**Portée**

Permet de vérifier si une entrée pour cette file d'attente existe également dans un service annuaire.

 L'attribut **Scope** n'est pas pris en charge sous z/OS

Pour une description complète de ces attributs, voir [Attributs des files d'attente](#).

**Concepts associés**

 **Files d'attente partagées**

Une file d'attente partagée et un type de file d'attente local. Les messages présents sur cette file d'attente peuvent faire l'objet d'un accès par un ou plusieurs gestionnaires de files d'attente se trouvant sur un sysplex.

[«Files d'attente de cluster», à la page 61](#)

Une file d'attente de cluster est une file d'attente hébergée par un gestionnaire de files d'attente de cluster et accessible aux autres gestionnaires de files d'attente dans le cluster.

[«Files d'attente de rebut», à la page 52](#)

La file d'attente de rebut (ou file d'attente des messages non livrés) est la file d'attente à laquelle des messages sont envoyés s'ils ne peuvent pas être acheminés vers la destination appropriée. Chaque gestionnaire de files d'attente a généralement une file d'attente de rebut.

**Tâches associées**

[Références relatives au développement d'applications](#)

**Référence associée**

[Les commandes MQSC](#)

[«Comparaison entre les files d'attente partagées et les files d'attente de cluster», à la page 62](#)

Ces informations vous permettent de comparer les files d'attente partagées et les files d'attente de cluster, et de sélectionner le type de file d'attente le plus approprié pour votre système.

**Files d'attente locales**

Les files d'attente de transmission, d'initialisation, de rebut, de commandes, par défaut, de canal et d'événements sont des types de files d'attente.

Une file d'attente est considérée par un programme comme étant *locale* si elle est détenue par le gestionnaire de files d'attente auquel le programme est connecté. Vous pouvez extraire et insérer des messages dans les files d'attente locales.

L'objet définition de file d'attente stocke les informations de définition de la file d'attente, ainsi que les messages physiques insérés dans la file d'attente.

Chaque gestionnaire de files d'attente peut comporter quelques files d'attente locales qu'il utilise à des fins spécifiques :

**Files d'attente de transmission**

Lorsqu'une application envoie un message à une file d'attente éloignée, le gestionnaire de files d'attente local stocke le message dans une file d'attente locale spéciale, appelée *file d'attente de*

*transmission*. Les applications peuvent placer les messages directement dans une file d'attente de transmission ou indirectement via une définition de file d'attente éloignée.

Lorsqu'un gestionnaire de files d'attente envoie des messages à un gestionnaire de files d'attente éloignées, il identifie la file d'attente de transmission dans l'ordre suivant :

1. La file d'attente de transmission nommée sur l'attribut XMITQ de la définition locale d'une file d'attente éloignée.
2. Une file d'attente de transmission portant le même nom que le gestionnaire de files d'attente éloignées. Cette valeur est la valeur par défaut sur l'attribut XMITQ de la définition locale d'une file d'attente éloignée.
3. La file d'attente de transmission nommée sur l'attribut DEFXMITQ du gestionnaire de files d'attente local.

Un *agent MCA* est un programme de canal qui est associé à la file d'attente de transmission et qui distribue le message à sa destination suivante. La destination suivante est le gestionnaire de files d'attente auquel le canal de messages est connecté. Il ne s'agit pas nécessairement du même gestionnaire de files d'attente que la destination finale du message. Une fois le message distribué à sa destination suivante, il est supprimé de la file d'attente de transmission. Le message a peut-être besoin de traverser plusieurs gestionnaires de files d'attente sur son parcours jusqu'à sa destination finale. Vous devez définir une file d'attente de transmission sur chaque gestionnaire de files d'attente du parcours, chacun stockant des messages en attente de transmission à la destination suivante. Une file d'attente de transmission normale contient les messages pour la destination suivante, même si ces messages ont différentes destinations possibles. Une file d'attente de transmission du cluster contient les messages destinés à plusieurs destinations. Le `correlID` de chaque message identifie le canal dans lequel le message est placé pour son transfert vers la destination suivante.

Vous pouvez définir plusieurs files d'attente de transmission au niveau d'un gestionnaire de files d'attente. Vous pouvez définir plusieurs file d'attente de transmission pour la même destination, chacune étant utilisée pour une classe de service différente. Par exemple, vous pouvez créer différentes files d'attente de transmission pour les messages de petite taille et de grande taille ayant la même destination. Vous pouvez ensuite transférer ces messages au moyen de différents canaux de messages, de sorte que les messages de grande taille ne bloquent pas les messages plus petits. Tous les messages pour les files d'attente de cluster ou les rubriques de cluster sont placés dans la file d'attente de transmission de cluster unique, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, par défaut. Vous pouvez éventuellement modifier la valeur par défaut et séparer le trafic des messages à destination des gestionnaires de files d'attente de cluster dans des files d'attente de transmission du cluster différentes. Si vous définissez l'attribut de gestionnaire de files d'attente `DEFCLXQ` sur `CHANNEL`, chaque canal émetteur de cluster crée une file d'attente de transmission de cluster distincte. Il est également possible de définir manuellement les files d'attente de transmission de cluster pour les canaux émetteurs de cluster à utiliser.

Les files d'attente de transmission peuvent déclencher un agent de canal de message pour envoyer les messages. Voir [Démarrage des applications IBM MQ en utilisant des déclencheurs](#).

 Sur IBM MQ for z/OS, si vous utilisez la mise en file d'attente intra-groupe, la file d'attente de transmission est traitée par un *agent de mise en file d'attente intra-groupe*. Une file d'attente de transmission est utilisée lors de l'utilisation de la mise en file d'attente intra-groupe sur IBM MQ for z/OS.

## Files d'attente d'initialisation

Une *file d'attente d'initialisation* est une file d'attente locale dans laquelle le gestionnaire de files d'attente insère un message de déclenchement lorsqu'un événement déclencheur se produit sur une file d'attente d'application.

Un événement déclencheur est un événement qui oblige un programme à démarrer le traitement d'une file d'attente. Par exemple, un événement peut être l'arrivée de plus de 10 messages. Pour plus d'informations sur le fonctionnement du déclenchement, voir [Démarrage des applications IBM MQ en utilisant des déclencheurs](#).

## File d'attente de rebut (de messages non livrés)

Une *file d'attente de rebut (de messages non livrés)* est une file d'attente locale dans laquelle le gestionnaire de files d'attente insère des messages qu'il ne peut pas distribuer.

Lorsque le gestionnaire de files d'attente insère un message dans la file d'attente de rebut, il ajoute un en-tête dans le message. Les informations d'en-tête incluent la raison pour laquelle le gestionnaire de files d'attente insère le message dans la file d'attente de rebut. Il contient également la destination du message d'origine, la date et l'heure auxquelles le gestionnaire de files d'attente a placé le message dans la file d'attente de rebut.

Les applications peuvent également utiliser la file d'attente pour les messages qu'elles ne peuvent pas distribuer. Pour plus d'informations, voir [Utilisation de la file d'attente de rebut \(de messages non livrés\)](#).

## File d'attente de commandes système

La *file d'attente de commandes système* est une file d'attente à laquelle les applications disposant de droits d'accès appropriés peuvent envoyer des commandes IBM MQ. Ces files d'attente reçoivent les commandes PCF, MQSC et CL, telles qu'elles sont prises en charge sur votre plateforme, de sorte que le gestionnaire de files d'attente les traite.

 Sous IBM MQ for z/OS, la file d'attente est appelée `SYSTEM.COMMAND.INPUT`; sur les autres plateformes, il est appelé `SYSTEM.ADMIN.COMMAND.QUEUE`. Les commandes acceptées varient selon la plateforme. Voir [Informations de référence sur les formats de commande programmables](#) pour plus de détails.

## Files d'attente par défaut du système

Les *files d'attente par défaut du système* contiennent les définitions initiales des files d'attente de votre système. Lors de la création d'une définition de file d'attente, le gestionnaire de files d'attente copie la définition à partir de la file d'attente par défaut appropriée du système. L'opération de création d'une définition de file d'attente est différente de l'opération de création d'une file d'attente dynamique. La définition de la file d'attente dynamique est basée sur la file d'attente modèle que vous choisissez comme modèle pour la file d'attente dynamique.

## Files d'attente d'événements

Les *files d'attente d'événements* stockent des messages d'événement. Ces messages sont signalés par le gestionnaire de files d'attente ou un canal.

## Files d'attente éloignées

Une file d'attente est considérée comme *éloignée* par un programme si elle est détenue par un gestionnaire de files d'attente différent de celui auquel le programme est connecté.

Lorsqu'une liaison de communication a été établie, un programme peut envoyer un message à une file d'attente éloignée. Un programme ne peut jamais extraire de message d'une file d'attente éloignée.

L'objet définition de file d'attente, créé lors de la définition d'une file d'attente éloignée, ne stocke que les informations nécessaires pour que le gestionnaire de files d'attente local recherche la file d'attente dans laquelle vous souhaitez placer votre message. Cet objet est appelé *définition locale d'une file d'attente éloignée*. Tous les attributs de la file d'attente éloignée sont stockés par le gestionnaire de files d'attente qui la détient, car elle est considérée comme file d'attente locale par ce gestionnaire de files d'attente.

Lors de l'ouverture d'une file d'attente éloignée, pour identifier cette dernière, vous devez indiquer l'un des éléments suivants :

- Le nom de la définition locale qui définit la file d'attente éloignée. Du point de vue d'une application, cette étape est la même que l'ouverture d'une file d'attente. Une application n'a pas besoin de savoir si une file d'attente est locale ou éloignée.

Pour créer une définition locale d'une file d'attente éloignée sur toutes les plateformes sauf IBM i, utilisez la commande `DEFINE QREMOTE`.

**IBM i** Sous IBM i, utilisez la commande `CRTMQMQ`.

- Le nom du gestionnaire de files d'attente éloignées et le nom de la file d'attente, tel qu'il est connu de ce gestionnaire de files d'attente éloignées.

Les définitions locales des files d'attente éloignées comportent trois attributs en sus des attributs communs décrits dans la section «Attributs des files d'attente», à la page 16. Il s'agit des trois attributs suivants :

**RemoteQName**

Nom sous lequel le gestionnaire de files d'attente propriétaire de la file d'attente connaît cette dernière.

**RemoteQMgrName**

Nom du gestionnaire de files d'attente propriétaire.

**XmitQName**

Nom de la file d'attente de transmission locale utilisée lors de l'acheminement des messages à d'autres gestionnaires de files d'attente.

Pour plus d'informations sur ces attributs, voir [Attributs des files d'attente](#).

Si vous utilisez l'appel MQINQ au niveau de la définition locale d'une file d'attente éloignée, le gestionnaire de files d'attente renvoie les attributs de la définition locale uniquement, à savoir le nom de la file d'attente éloignée, le nom du gestionnaire de files d'attente éloignées et le nom de la file d'attente de transmission, et non les attributs de la file d'attente locale correspondante du système éloigné.

Voir aussi [Files d'attente de transmission](#).

**Files d'attente alias**

Une *file d'attente alias* est un objet IBM MQ qui permet d'accéder à une autre file d'attente ou rubrique. Cela signifie que plusieurs programmes peuvent utiliser la même file d'attente en y accédant à l'aide de noms différents.

La file d'attente résultant de la résolution d'un nom d'alias, appelée file d'attente de base, peut être l'un des types de files d'attente pris en charge par la plateforme :

- Une file d'attente locale
- La définition locale d'une file d'attente éloignée.
- **z/OS** Une file d'attente partagée, qui est un type de file d'attente locale disponible uniquement sur IBM MQ for z/OS.
- Une file d'attente prédéfinie
- Une file d'attente dynamique

Un nom d'alias peut également correspondre à une rubrique. Si une application insère actuellement des messages dans une file d'attente, elle peut faire en sorte qu'ils soient publiés dans une rubrique en faisant du nom de la file d'attente un alias de la rubrique. Aucune modification du code d'application n'est nécessaire.

**Remarque :** Un alias ne peut pas se résoudre directement en un autre alias sur le même gestionnaire de files d'attente.

Exemple d'utilisation de files d'attente alias : un administrateur système accorde des droits d'accès différents au nom de file d'attente de base (c-à-d, la file d'attente à laquelle correspond l'alias) et au nom de file d'attente alias. En d'autres termes, un programme ou un utilisateur peut être autorisé à utiliser la file d'attente alias, et non la file d'attente de base.

Le cas échéant, des droits peuvent être définis de manière interdire les opérations d'insertion pour le nom d'alias mais les autoriser pour la file d'attente de base.

Dans certaines applications, l'utilisation de files d'attente alias signifie que les administrateurs système peuvent aisément modifier la définition d'un objet file d'attente alias sans avoir à modifier l'application.

IBM MQ effectue des contrôles d'autorisation au niveau du nom d'alias lorsque les programmes essaient d'utiliser ce nom. Il ne vérifie pas que le programme est autorisé à accéder au nom auquel correspond l'alias. Un programme peut donc être autorisé à accéder à un nom de file d'attente alias et non au nom de file d'attente résolu.

Outre les attributs de file d'attente généraux décrits dans la rubrique «Files d'attente», à la page 16, les files d'attente alias comportent un attribut **BaseQName**. Il s'agit du nom de la file d'attente de base auquel correspond le nom d'alias. Pour une description complète de cet attribut, voir BaseQName (MQCHAR48).

Les attributs *InhibitGet* et **InhibitPut** (voir «Files d'attente», à la page 16) des files d'attente alias font partie du nom d'alias. Par exemple, si le nom de file d'attente alias ALIAS1 correspond au nom de file d'attente de base BASE, les interdictions appliquées à ALIAS1 ont une incidence sur ALIAS1 uniquement, et BASE n'est pas interdit. Cependant, les interdictions appliquées à BASE ont également une incidence sur ALIAS1.

Les attributs *DefPriority* et **DefPersistence** font également partie du nom d'alias. Par conséquent, vous pouvez par exemple attribuer des priorités par défaut différentes à des alias différents de la même file d'attente de base. De même, vous pouvez modifier ces priorités sans avoir à modifier les applications qui utilisent les alias.

### **Files d'attente dynamiques et modèle**

Cette rubrique fournit un aperçu des files d'attente dynamiques, des propriétés des files d'attente dynamiques temporaires et permanentes, des utilisations de files d'attente dynamiques, quelques remarques sur l'utilisation des files d'attente dynamiques, ainsi que des files d'attente modèle.

Lorsqu'un programme d'application émet un appel MQOPEN pour ouvrir une file d'attente modèle, le gestionnaire de files d'attente crée de façon dynamique une file d'attente locale dont les attributs sont les mêmes que ceux de la file d'attente modèle. En fonction de la valeur de la zone *DefinitionType* de la file d'attente modèle, le gestionnaire de files d'attente crée une file d'attente dynamique temporaire ou permanente (voir [Création de files d'attente dynamiques](#)).

### **Propriétés des files d'attente dynamiques temporaires**

Les files d'attente dynamiques temporaires possèdent les propriétés suivantes :

-  Elles ne peuvent pas être des files d'attente partagées, accessibles à partir de gestionnaires de files d'attente d'un groupe de partage de files d'attente.

Notez que les files d'attente partagées et les groupes de partage de files d'attente sont disponibles uniquement dans IBM MQ for z/OS.

- Elles ne stockent que des messages non persistants.
- Elles sont irrécupérables.
- Elles sont supprimées lors du démarrage du gestionnaire de files d'attente.
- Elles sont supprimées lorsque l'application ayant émis l'appel MQOPEN qui a créé la file d'attente ferme la file d'attente ou prend fin.
  - Si la file d'attente contient des messages validés, ceux-ci sont supprimés.
  - S'il existe des appels MQGET, MQPUT ou MQPUT1 non validés en attente au niveau de la file d'attente à ce stade, la file d'attente est signalée comme étant supprimée logiquement et elle ne sera supprimée physiquement (une fois ces appels validés) que dans le cadre du processus de fermeture ou lorsque l'application prend fin.
  - Si la file d'attente est en cours d'utilisation (par l'application qui l'a créée ou par une autre application), la file d'attente est signalée comme étant supprimée logiquement et elle ne sera supprimée physiquement que lorsqu'elle est fermée par la dernière application qui l'utilise.
  - Les tentatives d'accès à une file d'attente supprimée logiquement (autres que la tentative de fermeture de la file d'attente) échouent en raison du code anomalie MQRC\_Q\_DELETED.

- Les options MQCO\_NONE, MQCO\_DELETE et MQCO\_DELETE\_PURGE sont toutes traités sous forme de MQCO\_NONE lorsqu'elles sont spécifiées dans un appel MQCLOSE pour l'appel MQOPEN correspondant qui a créé la file d'attente.

### Propriétés des files d'attente dynamiques permanentes

Les *files d'attente dynamiques permanentes* possèdent les propriétés suivantes :

- Elles stockent des messages persistants ou non persistants.
- Elles sont récupérables en cas d'incidents système.
- Elles sont supprimées lorsqu'une application (non nécessairement celle qui a émis l'appel MQOPEN qui a créé la file d'attente) ferme correctement la file d'attente à l'aide de l'option MQCO\_DELETE ou MQCO\_DELETE\_PURGE.
  - Une demande de fermeture à l'aide de l'option MQCO\_DELETE échoue si la file d'attente contient toujours des messages (validés ou non validés). Une demande de fermeture à l'aide de l'option MQCO\_DELETE\_PURGE aboutit même si la file d'attente contient des messages validés (les messages en cours de suppression dans le cadre de la fermeture), mais échoue s'il existe des appels MQGET, MQPUT ou MQPUT1 en attente au niveau de la file d'attente.
  - Si la demande de suppression aboutit et que la file d'attente est en cours d'utilisation (par l'application qui l'a créée ou par une autre application), la file d'attente est signalée comme étant supprimée logiquement et elle ne sera supprimée physiquement que lorsqu'elle est fermée par la dernière application qui l'utilise.
- Elles ne sont pas supprimées si elles sont fermées par une application non autorisée à supprimer la file d'attente, sauf si l'application qui les ferme a émis l'appel MQOPEN qui a créé la file d'attente. Les vérifications d'autorisation s'effectuent au niveau de l'ID utilisateur (ou un ID utilisateur secondaire si l'option MQOO\_ALTERNATE\_USER\_AUTHORITY a été spécifiée) qui a servi à valider l'appel MQOPEN correspondant.
- Elles peuvent être supprimées de la même façon qu'une file d'attente normale.

### Utilisations des files d'attente dynamiques

Vous pouvez utiliser des files d'attente dynamiques pour :

- Les applications qui n'exigent pas la conservation des files d'attente après l'arrêt de l'application.
- Les applications demandant des réponses aux messages à traiter par une autre application. Les applications de ce type peuvent créer de façon dynamique une file d'attente de réponses en ouvrant une file d'attente modèle. Par exemple, une application client peut :
  1. Créer une file d'attente dynamique.
  2. Fournir son nom dans la zone **ReplyToQ** de la structure de descripteur du message de demande.
  3. Placer la demande dans une file d'attente traitée par un serveur.

Le serveur peut ensuite placer le message de réponse dans la file d'attente de réponses. Pour terminer, le client peut traiter la réponse et fermer la file d'attente de réponses à l'aide de l'option de suppression.

### Remarques sur l'utilisation des files d'attente dynamiques

Prenez en considération les points suivants lors de l'utilisation des files d'attente dynamiques :

- Dans un modèle client-serveur, chaque client doit créer et utiliser sa propre file d'attente de réponses dynamique. Si une file d'attente de réponses dynamique est partagée entre plusieurs clients, la suppression de la file d'attente de réponses peut être retardée car la file d'attente contient des activités non validées en attente ou qu'elle est en cours d'utilisation par un autre client. En outre, il se peut que la file d'attente soit signalée comme étant supprimée logiquement et inaccessible pour les demandes d'API ultérieures (autres que MQCLOSE).
- Si votre environnement d'application exige que les files d'attente dynamiques soient partagées entre les applications, assurez-vous que la file d'attente n'est fermée (à l'aide de l'option de suppression) que lorsque toutes les activités au niveau de la file d'attente ont été validées. Cette opération doit être effectuée par le dernier utilisateur. Cela garantit que la suppression de la file

d'attente n'est pas retardée et permet de réduire la période pendant laquelle la file d'attente est inaccessible car elle a été signalée comme étant supprimée logiquement.

## Files d'attente modèle

Une *file d'attente modèle* est un modèle de définition de file d'attente utilisé lors de la création d'une file d'attente dynamique.

Vous pouvez créer une file d'attente locale de façon dynamique à partir d'un programme IBM MQ en désignant la file d'attente modèle que vous souhaitez utiliser comme modèle pour les attributs de file d'attente. A ce stade, vous pouvez modifier quelques attributs de la nouvelle file d'attente. Cependant, vous ne pouvez pas modifier la zone **DefinitionType**. Si, par exemple, vous avez besoin d'une file d'attente permanente, sélectionnez une file d'attente modèle dont le type de définition est paramétré sur permanente. Certaines applications conversationnelles peuvent utiliser des files d'attente dynamiques pour stocker les réponses à leurs requêtes, car elles n'ont probablement pas besoin de maintenir ces files d'attente une fois qu'elles ont traité les réponses.

Vous indiquez le nom d'une file d'attente modèle dans le *descripteur d'objet* (MQOD) de votre appel MQOPEN. A l'aide des attributs de la file d'attente modèle, le gestionnaire de files d'attente crée de façon dynamique une file d'attente locale.

Vous pouvez spécifier un nom (complet) pour la file d'attente dynamique ou la racine d'un nom (par exemple, ABC) et laisser le gestionnaire de files d'attente y ajouter une partie unique, ou vous pouvez demander au gestionnaire de files d'attente d'attribuer un nom unique complet. Si le gestionnaire de files d'attente attribue le nom, il le place dans la structure MQOD.

Vous ne pouvez pas émettre directement un appel MQPUT1 à une file d'attente modèle, mais vous pouvez émettre un appel MQPUT1 à la file d'attente dynamique qui a été créée via l'ouverture d'une file d'attente modèle.

MQSET et MQINQ ne peuvent pas être exécutés sur une file d'attente modèle. L'ouverture d'une file d'attente modèle avec MQOO\_INQUIRE ou MQOO\_SET entraîne l'exécution d'appels MQINQ et MQSET ultérieurs sur la file d'attente créée dynamiquement.

Les attributs d'une file d'attente modèle sont un sous-ensemble de ceux d'une file d'attente locale. Pour une description complète, voir [Attributs des files d'attente](#).

## Méthodes de définition des files d'attente

Vous pouvez créer une file d'attente à l'aide de commandes IBM MQ (MQSC), de commandes PCF ou des interfaces spécifiques aux plateformes.

Vous pouvez définir des files d'attente vers IBM MQ à l'aide de la commande MQSC [DEFINE](#) ou de la commande PCF de [création de file d'attente \(Create Queue\)](#). Les commandes indiquent le type de file d'attente et ses attributs. Par exemple, un objet file d'attente locale comporte des attributs indiquant ce qui se produit lorsque les applications font référence à cette file d'attente dans les appels MQI. Voici des exemples d'attributs :

- Attribut indiquant si les applications peuvent extraire des messages de la file d'attente (GET activé)
- Attribut indiquant si les applications peuvent placer des messages dans la file d'attente (PUT activé)
- Attribut indiquant si l'accès à la file d'attente est réservé exclusivement à une application ou s'il est partagé entre les applications
- Nombre maximal de messages pouvant être stockés simultanément dans la file d'attente (longueur maximale de la file d'attente)
- Longueur maximale des messages pouvant être placés dans la file d'attente

Il existe également diverses interfaces spécifiques à la plateforme que vous pouvez utiliser pour définir des files d'attente. Pour des liens vers des informations supplémentaires, reportez-vous aux *informations connexes*.

## Concepts associés

[Administration à l'aide de commandes MQSC](#)

[Utilisation des files d'attente dans IBM MQ Console](#)

### **Tâches associées**

[Administration de l'automatisation à l'aide de commandes PCF](#)

[Création et configuration de gestionnaires de files d'attente et d'objets avec MQ Explorer](#)

[Gestion d'IBM MQ for IBM i à l'aide de commandes CL](#)

[Emission de commandes sur IBM MQ for z/OS](#)

### **Files d'attente utilisées à des fins spécifiques par IBM MQ**

IBM MQ utilise certaines files d'attente locales pour certains buts liés à ses opérations.

Vous devez définir ces files d'attente avant qu'IBM MQ puisse les utiliser.

#### **Files d'attente d'initialisation**

Les files d'attente d'initialisation sont des files d'attente utilisées pour le déclenchement. Un gestionnaire de files d'attente place un message de déclenchement dans une file d'attente d'initialisation lorsqu'un événement déclencheur se produit. Un événement déclencheur est une combinaison logique de conditions qui est détectée par un gestionnaire de files d'attente. Par exemple, un événement déclencheur peut être généré lorsque le nombre de messages dans une file d'attente atteint une longueur prédéfinie. Cet événement fait en sorte que le gestionnaire de files d'attente place un message de déclenchement dans une file d'attente d'initialisation indiquée. Ce message de déclenchement est extrait par un *moniteur de déclenchement*, qui correspond à une application spéciale surveillant une file d'attente d'initialisation. Le moniteur de déclenchement démarre ensuite le programme d'application qui a été indiqué dans le message de déclenchement.

Si un gestionnaire de files d'attente doit utiliser le déclenchement, au moins une file d'attente d'initialisation doit être définie pour ce gestionnaire de files d'attente. Voir [Gestion des objets pour le déclenchement](#), [runmqtrm](#) et [Démarrage des applications IBM MQ en utilisant des déclencheurs](#).

#### **Files d'attente de transmission**

Les files d'attente de transmission sont des files d'attente qui stockent provisoirement des messages destinés à un gestionnaire de files d'attente éloignées. Vous devez définir au moins une file d'attente de transmission pour chaque gestionnaire de files d'attente éloignées auquel le gestionnaire de files d'attente local doit envoyer directement des messages. Ces files d'attente sont également utilisées pour l'administration à distance. Voir [Administration à distance à partir d'un gestionnaire de files d'attente local](#). Pour des informations sur l'utilisation des files d'attente de transmission pour la mise en file d'attente répartie, voir [Techniques de distribution répartie d'IBM MQ](#).

Chaque gestionnaire de files d'attente peut posséder une file d'attente de transmission par défaut. Si un gestionnaire de files d'attente ne faisant pas partie d'un cluster place un message dans une file d'attente éloignée, l'action par défaut consiste à utiliser la file d'attente de transmission par défaut. S'il existe une file d'attente de transmission dotée du même nom que le gestionnaire de files d'attente de destination, le message est placé dans cette file d'attente de transmission. S'il existe une définition d'alias de gestionnaire de files d'attente, dans lequel le paramètre **RQMNAME** correspond au gestionnaire de files d'attente de destination, et si le paramètre **XMITQ** est spécifié, le message est placé dans la file d'attente de transmission appelée par **XMITQ**. S'il n'existe aucun paramètre **XMITQ**, le message est placé dans la file d'attente locale nommée dans le message.

#### **Files d'attente de transmission de cluster**

Chaque gestionnaire de files d'attente au sein d'un cluster comporte une file d'attente de transmission appelée `SYSTEM.CLUSTER.TRANSMIT.QUEUE`, et file d'attente de transmission de cluster modèle, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Les définitions de ces files d'attente sont créées par défaut lorsque vous définissez un gestionnaire de files d'attente. Si l'attribut de gestionnaire de files d'attente, **DEFCLXQ**, est défini sur `CHANNEL`, une file d'attente de transmission de cluster dynamique permanente est automatiquement créée pour chaque canal émetteur de cluster qui est créé. Les files d'attente sont appelées `SYSTEM.CLUSTER.TRANSMIT.ChanneName`. Vous pouvez également définir les files d'attente de transmission de cluster manuellement.

Un gestionnaire de files d'attente faisant partie du cluster envoie des messages d'une de ces files d'attente aux autres gestionnaires de files d'attente du même cluster.

Lors de la résolution de nom, une file d'attente de transmission de cluster est prioritaire sur la file d'attente de transmission par défaut, et une file d'attente de transmission de cluster spécifique est prioritaire sur SYSTEM . CLUSTER . TRANSMIT . QUEUE.

### **Files d'attente de rebut**

Une file d'attente de rebut (messages non livrés) est une file d'attente qui stocke les messages qui ne peuvent pas être acheminés vers leurs destinations correctes. Un message ne peut pas être acheminé lorsque, par exemple, la file d'attente de destination est saturée. La file d'attente de rebut fournie est appelée SYSTEM . DEAD . LETTER . QUEUE.

Pour la mise en file d'attente répartie, définissez une file d'attente de rebut dans chaque gestionnaire de files d'attente impliqué.

### **Files d'attente de commandes**

La file d'attente de commandes, SYSTEM . ADMIN . COMMAND . QUEUE, est une file d'attente locale à laquelle les applications disposant de droits d'accès appropriés peuvent envoyer des commandes MQSC à des fins de traitement. Ces commandes sont ensuite extraites par un composant IBM MQ appelé serveur de commandes. Le serveur de commandes valide les commandes, transmet les commandes valides à des fins de traitement par le gestionnaire de files d'attente, puis renvoie les réponses à la file d'attente de réponses appropriée.

Une file d'attente de commandes est créée automatiquement pour chaque gestionnaire de files d'attente lorsque ce dernier est créé.

### **Files d'attente de réponses**

Lorsqu'une application envoie un message de demande, l'application qui reçoit le message peut renvoyer un message de réponse à l'application émettrice. Ce message est placé dans une file d'attente, appelée file d'attente de réponses, qui est généralement une file d'attente locale de l'application émettrice. Le nom de la file d'attente de réponses est indiqué par l'application émettrice dans le cadre du descripteur de message.

### **Files d'attente d'événements**

Les événements outils permettent de surveiller les gestionnaires de files d'attente indépendamment des applications MQI.

Lorsqu'un événement outil se produit, le gestionnaire de files d'attente place un message d'événement dans une file d'attente d'événements. Ce message peut ensuite être lu par une application de surveillance qui peut informer un administrateur ou déclencher une action corrective si l'événement indique un problème.

**Remarque :** Les événements déclencheurs sont assez différents des événements outils. Les événements déclencheurs ne sont pas occasionnés par les mêmes conditions et ne génèrent pas de messages d'événement.

Pour plus d'informations sur les événements outils, voir [Événements outils](#).

## **Gestionnaires de files d'attente IBM MQ**

Présentation des *gestionnaires de files d'attente* et des services de mise en file d'attente qu'ils fournissent aux applications.

Un programme doit disposer d'une connexion au gestionnaire de files d'attente avant qu'il puisse utiliser les services de ce gestionnaire de files d'attente. Un programme peut établir cette connexion de façon explicite (à l'aide de l'appel MQCONNX) ou la connexion peut être établie de façon implicite (en fonction de la plateforme et de l'environnement dans lequel le programme est en cours d'exécution).

Les gestionnaires de files d'attente fournissent des services de mise en file d'attente aux applications et gèrent les files d'attente qui leur appartiennent. Un gestionnaire de files d'attente vérifie les actions suivantes :

- Les attributs d'objet sont modifiés selon les commandes reçues.
- Des événements spéciaux, tels que les événements déclencheurs ou les événements outils, sont générés lorsque les conditions appropriées sont remplies.

- Les messages sont placés dans la file d'attente correcte, à la demande de l'application qui établit l'appel MQPUT. L'application est informée si cela n'est pas possible et un code anomalie approprié est généré.

Chaque file d'attente fait partie d'un gestionnaire de files d'attente unique et est considérée comme étant une *file d'attente locale* de ce gestionnaire de files d'attente. Le gestionnaire de files d'attente auquel une application est connectée est considéré comme étant le *gestionnaire de files d'attente local* de cette application. Pour l'application, les files d'attente faisant partie de son gestionnaire de files d'attente local sont des files d'attente locales.

Une *file d'attente éloignée* est une file d'attente appartenant à un autre gestionnaire de files d'attente. Un *gestionnaire de files d'attente éloignées* est un gestionnaire de files d'attente autre que le gestionnaire de files d'attente local. Un gestionnaire de files d'attente éloignées peut résider sur une machine distante du réseau ou sur la même machine que le gestionnaire de files d'attente local. IBM MQ prend en charge plusieurs gestionnaires de files d'attente sur une même machine.

Un objet gestionnaire de files d'attente peut être utilisé dans certains appels MQI. Par exemple, vous pouvez visualiser les attributs de l'objet gestionnaire de files d'attente à l'aide de l'appel MQI MQINQ.

## Attributs des gestionnaires de files d'attente

Chaque gestionnaire de files d'attente est associé à un ensemble d'attributs (ou de propriétés) définissant ses caractéristiques. Certains attributs d'un gestionnaire de files d'attente sont fixes lorsque celui-ci est créé ; vous pouvez en modifier d'autres à l'aide des commandes IBM MQ. Vous pouvez visualiser les valeurs de tous les attributs, à l'exception de ceux utilisés pour le chiffrement Transport Sockets Layer (TLS), à l'aide de l'appel MQINQ.

Les attributs fixes sont les suivants :

- Nom du gestionnaire de files d'attente
- Plateforme sur laquelle s'exécute le gestionnaire de files d'attente (par exemple, Windows)
- Niveau des commandes de contrôle système prises en charge par le gestionnaire de files d'attente
- Priorité maximale pouvant être affectée aux messages traités par le gestionnaire de files d'attente
- Nom de la file d'attente à laquelle les programmes peuvent envoyer des commandes IBM MQ
- Longueur maximale des messages que peut traiter le gestionnaire de files d'attente  (fixe uniquement dans IBM MQ for z/OS)
- Prise en charge ou non de point de synchronisation par le gestionnaire de files d'attente lorsque des programmes placent et extraient des messages

Les attributs *modifiables* sont les suivants :

- Description du gestionnaire de files d'attente
- Identificateur du jeu de caractères utilisé par le gestionnaire de files d'attente pour les chaînes de caractères lorsqu'il traite les appels MQI
- Intervalle de temps utilisé par le gestionnaire de files d'attente pour restreindre le nombre de messages de déclenchement
-  Intervalle de temps qu'utilise le gestionnaire de files d'attente pour déterminer la fréquence à laquelle il doit rechercher les messages expirés dans les files d'attente (IBM MQ for z/OS uniquement)
- Nom de la file d'attente de rebut (messages non livrés) du gestionnaire de files d'attente
- Nom de la file d'attente de transmission par défaut du gestionnaire de files d'attente
- Nombre maximal de descripteurs ouverts pour une connexion
- Activation et désactivation de diverses catégories de rapport d'événement
- Nombre maximal de messages non validés au sein d'une unité d'oeuvre

## Gestionnaires de files d'attente et gestion de la charge de travail

Vous pouvez configurer un cluster de gestionnaires de files d'attente comportant plusieurs définitions pour la même file d'attente (par exemple, les gestionnaires de files d'attente du cluster peuvent être des clones les uns des autres). Les messages d'une file d'attente particulière peuvent être gérés par n'importe quel gestionnaire de files d'attente hébergeant une instance de la file d'attente. Un algorithme de gestion de la charge de travail détermine le gestionnaire de files d'attente qui traite le message et répartit ainsi la charge de travail entre vos gestionnaires de files d'attente ; pour plus d'informations, voir [>Algorithme de gestion de la charge de travail des clusters](#).

## Définitions de processus

Les objets définition de processus permettent le démarrage des applications sans l'intervention de l'opérateur, en définissant les attributs de l'application à des fins d'utilisation par le gestionnaire de files d'attente.

L'objet définition de processus définit une application qui démarre en réponse à un événement déclencheur sur un gestionnaire de files d'attente IBM MQ. Les attributs de définition de processus incluent l'ID application, le type d'application et des données spécifiques à l'application. Pour plus d'informations, voir *Files d'attente d'initialisation* dans la rubrique [«Files d'attente utilisées à des fins spécifiques par IBM MQ»](#), à la page 24.

Pour permettre à une application de démarrer sans l'intervention de l'opérateur, comme décrit dans [Démarrage des applications IBM MQ en utilisant des déclencheurs](#), les attributs de l'application doivent être connus du gestionnaire de files d'attente. Ces attributs sont définis dans un *objet définition de processus*.

L'attribut **ProcessName** est fixe lorsque l'objet est créé. Toutefois, vous pouvez modifier les autres attributs à l'aide des commandes IBM MQ.  Vous pouvez également, sous z/OS, utiliser les panneaux d'opérations et de contrôle IBM MQ for z/OS.

Vous pouvez interroger les valeurs de tous les attributs à l'aide de [MQINQ - Consultation des attributs d'un objet](#).

### Référence associée

[Attributs des définitions de processus](#)

## Listes de noms

*namelist* est un objet IBM MQ contenant une liste de noms de cluster, de noms de file d'attente ou de noms d'objet d'information d'authentification. Dans un cluster, elle permet d'identifier la liste des clusters pour lesquels le gestionnaire de files d'attente détient les référentiels.

Une liste de noms est un objet IBM MQ qui contient une liste d'autres objets IBM MQ. En général, les listes de noms sont utilisées par des applications telles que les moniteurs de déclenchement, où ces listes de noms sont alors utilisées pour identifier un groupe de files d'attente. L'utilisation d'une liste de noms présente un avantage : elle est gérée indépendamment des applications ; elle peut être mise à jour sans arrêter les applications qui l'utilisent. De même, si une application échoue, la liste de noms n'est pas concernée et les autres applications peuvent continuer de l'utiliser.

Les listes de noms sont également utilisées avec les clusters de gestionnaires de files d'attente pour gérer une liste de clusters auxquels se réfèrent plusieurs objets IBM MQ.

Vous pouvez définir et modifier des listes de noms à l'aide de commandes MQSC.  Vous pouvez également, sous z/OS, utiliser les panneaux d'opérations et de contrôle IBM MQ for z/OS.

Les programmes peuvent utiliser l'interface MQI pour identifier les files d'attente qui sont incluses dans ces listes de noms. L'organisation des listes de noms incombe au concepteur d'application et à l'administrateur système.

### Référence associée

[Attributs des listes de noms](#)

## Objets d'informations d'authentification

Un objet d'informations d'authentification fournit les définitions requises pour la vérification de la révocation de certificat.

L'objet d'information d'authentification du gestionnaire de files d'attente fait partie de la prise en charge de Transport Layer Security (TLS) par IBM MQ. Il fournit les informations nécessaires pour la vérification des certificats révoqués. Les autorités de certification révoquent les certificats qui ne sont plus sécurisés.

Vous pouvez utiliser la commande MQSC **DEFINE AUTHINFO** pour définir un objet d'informations d'authentification. Pour plus d'informations sur les attributs des objets d'informations d'authentification, voir **DEFINE AUTHINFO**.

Vous pouvez utiliser les commandes de contrôle IBM MQ suivantes avec un objet d'informations d'authentification :

- **setmqaut** (octroi ou révocation des droits d'accès)
- **dspmqaut** (affichage des autorisations des objets)
- **dmpmqaut** (vidage des autorisations)
- **rcrmqobj** (recréation d'objet)
- **rcdmqimg** (enregistrement d'image de support)
- **dspmqls** (affichage des noms de fichier)

Pour une présentation du protocole TLS, ainsi que de l'utilisation des objets d'informations d'authentification, voir [Protocoles de sécurité TLS dans IBM MQ](#).

### Concepts associés

[Concepts TLS \(Transport Layer Security\)](#)

## Objets d'information de communication

IBM MQ Multicast fournit une messagerie multidiffusion fiable, à haute distribution et à faible latence. Un objet d'information de communication (COMMINFO) est nécessaire pour utiliser la transmission Multicast.

Un objet COMMINFO est un objet IBM MQ contenant les attributs associés à la transmission multidiffusion. Pour plus d'informations sur ces attributs, voir [DEFINE COMMINFO](#). Pour plus d'informations sur la création d'un objet COMMINFO, voir [Guide d'initiation à la multidiffusion](#).

### Concepts associés

«Multidiffusion IBM MQ», à la page 114

IBM MQ Multicast fournit une messagerie multidiffusion fiable, à haute distribution et à faible latence.

## Canaux

Un *canal* est une liaison de communication logique, utilisée par les gestionnaires de files d'attente réparties, entre un serveur IBM MQ MQI client et un serveur IBM MQ ou entre deux serveurs IBM MQ.

Les canaux sont des objets qui fournissent un chemin de communication d'un gestionnaire de files d'attente vers un autre. Les canaux sont utilisés pour déplacer des messages d'un gestionnaire de files d'attente vers un autre et ils protègent les applications des protocoles de communication sous-jacents. Les gestionnaires de files d'attente peuvent résider sur la même plateforme ou sur des plateformes différentes. Les messages envoyés peuvent provenir de nombreux endroits :

- programmes d'application écrits par l'utilisateur qui transfèrent des données d'un noeud vers un autre,
- applications d'utilisateur écrites par l'utilisateur qui utilisent des commandes PCF ou WebSphere MQ Administration Interface,
- IBM MQ Explorer,

- gestionnaires de files d'attente qui envoient des messages d'événement d'instrumentation à un autre gestionnaire de files d'attente,
- gestionnaires de files d'attente qui envoient des commandes d'administration distantes à un autre gestionnaire de files d'attente. Par exemple, via l'utilisation de commandes MQSC ou de l'administrative REST API.

Un canal comporte deux définitions : une à chaque extrémité de la connexion. Pour que les gestionnaires de communication puissent communiquer ensemble, vous devez définir un objet canal sur le gestionnaire de files d'attente qui envoie les messages et un autre (complémentaire) sur le gestionnaire de files d'attente qui reçoit les messages. Le même *nom de canal* doit être utilisé à chaque extrémité de la connexion, et le *type de canal* utilisé doit être compatible.

Il existe trois catégories de canaux dans IBM MQ, chacune comportant des types de canaux différents :

- Les canaux de messages, qui sont unidirectionnels et qui transfèrent les messages d'un gestionnaire de files d'attente à un autre.
- Les canaux MQI qui sont bidirectionnels et qui transfèrent les appels MQI d'un IBM MQ MQI client vers un gestionnaire de files d'attente, ainsi que les réponses d'un gestionnaire de files d'attente vers un client IBM MQ.
- Les canaux AMQP, qui sont bidirectionnels et qui connectent un client AMQP à un gestionnaire de files d'attente sur un serveur. IBM MQ utilise des canaux AMQP pour transférer des appels et des réponses AMQP entre des gestionnaires de files d'attente et des applications AMQP.

## Canaux de transmission de messages

Un canal de transmission de message a pour but de transférer des messages d'un gestionnaire de files d'attente vers un autre. Les canaux de transmission de message ne sont pas requis par l'environnement client-serveur.

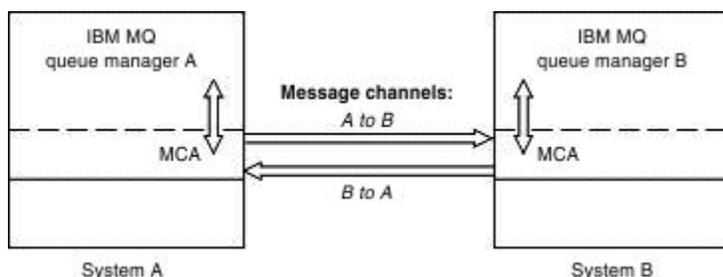


Figure 2. Canaux de transmission de message entre deux gestionnaires de files d'attente

Un canal de transmission de message est une liaison unidirectionnelle. Si vous souhaitez qu'un gestionnaire de files d'attente éloigné réponde aux messages envoyés par un gestionnaire de files d'attente local, vous devez configurer un second canal pour le renvoi des réponses au gestionnaire de files d'attente local.

Un canal de transmission de messages connecte deux gestionnaires de files d'attente grâce à des *Agents MCA*. Un agent MCA se trouve à chaque extrémité d'un canal. Vous pouvez autoriser un agent MCA à transférer des messages via plusieurs unités d'exécution. Ce processus est connu sous le nom de *principe du pipeline*. Le principe du pipeline permet à l'agent MCA de transférer des messages plus efficacement, améliorant ainsi les performances de canal. Pour plus d'informations sur le principe du pipeline, voir [Attributs des canaux](#).

Pour plus d'informations sur les canaux, voir [Structures de données et appels d'exit de canal](#) et «Composants de la mise en file d'attente répartie», à la page 49.

## Canaux MQI

Un canal MQI permet de connecter un IBM MQ MQI client à un gestionnaire de files d'attente sur une machine serveur et est établi lorsque vous émettez un appel MQCONN ou MQCONNX à partir d'une application IBM MQ MQI client.

Il s'agit d'une liaison bidirectionnelle qui sert à transférer uniquement des appels MQI et des réponses, y compris des appels MQPUT contenant des données de message et des appels MQGET donnant lieu au renvoi de données de message. Différentes méthodes permettent de créer et d'utiliser les définitions de canal (voir [Définition de canaux MQI](#)).

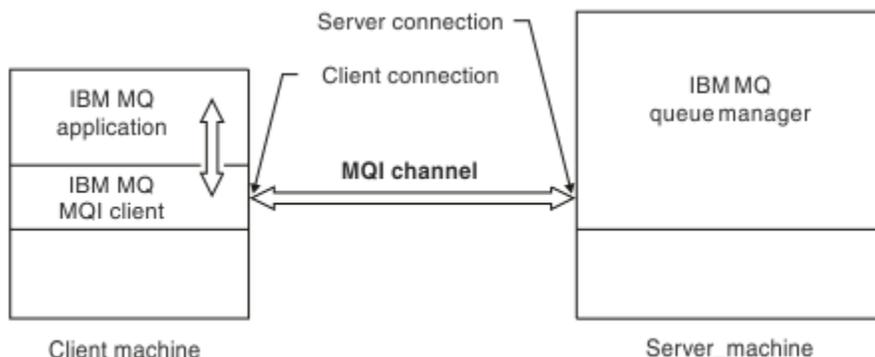


Figure 3. Connexion client et connexion serveur sur un canal MQI

**z/OS** Un canal MQI permet de connecter un client à un gestionnaire de files d'attente unique ou à un gestionnaire de files d'attente faisant partie d'un groupe de partage de files d'attente (voir [Connexion d'un client à un groupe de partage de files d'attente](#)).

Il existe deux types de canal pour les définitions de canal MQI. Ils définissent le canal MQI bidirectionnel.

#### Canal de connexion client

Ce type est destiné au IBM MQ MQI client.

#### Canal de connexion serveur

Ce type est destiné au serveur qui exécute le gestionnaire de files d'attente, avec lequel l'application IBM MQ en cours d'exécution dans un environnement IBM MQ MQI client, doit communiquer.

### Canaux AMQP

**Multi**

Il n'existe qu'un type de canal AMQP.

Vous utilisez ce canal pour connecter une application de messagerie AMQP à un gestionnaire de files d'attente, pour permettre à l'application d'échanger des messages avec les applications IBM MQ. Un canal AMQP permet de développer une application à l'aide de MQ Light, puis de la déployer comme application d'entreprise pour profiter des utilitaires de niveau entreprise fournis par IBM MQ.

### Définitions de canal

Voir [«Définitions de canal»](#), à la page 31 pour une description de chaque type de canal.

#### Concepts associés

[«Mise en file d'attente répartie et clusters»](#), à la page 45

La mise en file d'attente répartie permet d'envoyer des messages d'un gestionnaire de files d'attente à un autre. Le gestionnaire de files d'attente récepteur peut résider sur la même machine ou sur une autre, à proximité ou à l'autre bout du monde. Il peut s'exécuter sur la même plateforme que le gestionnaire de files d'attente local ou se trouver sur n'importe laquelle des plateformes prises en charge par IBM MQ. Vous pouvez définir manuellement toutes les connexions dans un environnement de mise en file d'attente répartie, ou créer un cluster et laisser IBM MQ définir la plupart des informations de connexion automatiquement.

[Présentation de l'interface MQI \(Message Queue Interface\)](#)

#### Tâches associées

[Administration des objets IBM MQ éloignés](#)

Arrêt des canaux MQI

### Référence associée

Structures de données et appels d'exit de canal

«Communications», à la page 34

Les IBM MQ MQI clients utilisent des canaux MQI pour communiquer avec le serveur.

### Définitions de canal

Tableaux décrivant les différents types de canaux de transmission de messages utilisés par IBM MQ.

Pour désigner les canaux de transmission de messages, le terme canal est souvent utilisé comme synonyme de définition de canal. Le contexte permet généralement de savoir si l'on parle d'un canal complet, avec deux extrémités, ou d'une définition de canal avec une seule extrémité.

### Canaux de transmission de messages

Les définitions de canaux de transmission de messages peuvent être de différents types :

Type de définition de canaux de transmission de messages	Description
Emetteur	Un canal émetteur est un canal de transmission de messages utilisé par le gestionnaire de files d'attente pour envoyer des messages à d'autres gestionnaires de files d'attente. Pour envoyer des messages via un canal émetteur, vous devez aussi créer, sur l'autre gestionnaire de files d'attente, un canal récepteur portant le même nom que le canal émetteur. Vous pouvez également utiliser des canaux émetteurs avec des canaux demandeurs si vous mettez en place un mécanisme de procédure appelée (callback).
serveur	Un canal serveur est un canal de transmission de messages utilisé par le gestionnaire de files d'attente pour envoyer des messages à d'autres gestionnaires de files d'attente. Pour envoyer des messages via un canal serveur, vous devez aussi créer, sur l'autre gestionnaire de files d'attente, un canal récepteur portant le même nom que le canal serveur. Vous pouvez également utiliser des canaux serveur avec des canaux demandeur. Dans ce cas, la définition du canal demandeur à l'autre extrémité du canal demande à la définition du canal serveur de démarrer. Le serveur envoie des messages au demandeur. Le serveur peut également initialiser la communication s'il connaît le nom de connexion du canal partenaire.
Récepteur	Un canal récepteur est un canal de transmission de messages utilisé par le gestionnaire de files d'attente pour recevoir des messages en provenance d'autres gestionnaires de files d'attente. Pour recevoir des messages via un canal récepteur, vous devez aussi créer, sur l'autre gestionnaire de files d'attente, un canal émetteur ou serveur portant le même nom que le canal récepteur.

Type de définition de canaux de transmission de messages	Description
Demandeur	<p>Un canal demandeur est un canal de transmission de messages utilisé par le gestionnaire de files d'attente pour recevoir des messages d'autres gestionnaires de files d'attente. Un canal demandeur peut demander le canal partenaire défini sur l'extrémité distante pour démarrer. Si le canal partenaire est un canal serveur, le canal serveur accepte la demande de démarrage et commence à envoyer des messages depuis la file d'attente de transmission identifiée dans la définition de canal serveur au canal demandeur. Si le canal partenaire est un canal émetteur, le canal émetteur accepte la demande de démarrage, puis ferme la connexion au demandeur. Le canal émetteur démarre, négocie une session avec le canal demandeur partenaire, puis commence à envoyer des messages depuis la file d'attente de transmission identifiée dans la définition du canal émetteur. Ce dernier cas fournit essentiellement un mécanisme de rappel car le canal demandeur demande au canal émetteur de rappeler.</p>
Emetteur de cluster	<p>Une définition de canal émetteur de cluster (CLUSDR) définit l'extrémité émettrice d'un canal, sur laquelle un gestionnaire de files d'attente du cluster peut envoyer des informations relatives au cluster à l'un des référentiels complets. Le canal émetteur de cluster permet d'informer le référentiel en cas de modification du statut du gestionnaire de files d'attente, par exemple en cas d'ajout ou de suppression d'une file d'attente. Il permet également de transmettre des messages. Les gestionnaires de files d'attente de référentiel complet eux-mêmes disposent de canaux émetteurs de cluster qui pointent l'un vers l'autre. Ils leur permettent de se communiquer mutuellement les modifications du statut du cluster. Il n'est pas très important de savoir vers quel référentiel complet pointe une définition de canal CLUSSDR du gestionnaire de files d'attente. Une fois le premier contact établi, un plus grand nombre d'objets gestionnaire de files d'attente du cluster est défini automatiquement selon les besoins, de sorte que le gestionnaire de files d'attente puisse envoyer des informations relatives au cluster à tous les référentiels complets, et des messages à tous les gestionnaires de files d'attente.</p>

Type de définition de canaux de transmission de messages	Description
Récepteur de cluster	Une définition de canal récepteur de cluster (CLUSRCVR) définit l'extrémité réceptrice d'un canal, sur laquelle un gestionnaire de files d'attente du cluster peut recevoir des messages d'autres gestionnaires de files d'attente du cluster. Un canal récepteur de cluster peut également transmettre des informations relatives au cluster destinées au référentiel. En définissant le canal récepteur de cluster, le gestionnaire de files d'attente indique aux autres gestionnaires qu'il est disponible pour recevoir des messages. Vous devez disposer d'au moins un canal récepteur de cluster pour chaque gestionnaire de files d'attente du cluster.

Pour chaque canal, vous devez définir les deux extrémités afin de disposer d'une définition de canal pour chaque extrémité. Les deux extrémités du canal doivent avoir des types compatibles.

Vous pouvez combiner les définitions de canal de la façon suivante :

- Emetteur-Récepteur
- Serveur-Récepteur
- Demandeur-Serveur
- Demandeur-Emetteur (rappel)
- Emetteur de cluster-Récepteur de cluster

## Agents MCA

Chaque définition de canal créée appartient à un gestionnaire de files d'attente spécifique. Un gestionnaire peut comporter plusieurs canaux d'un type identique ou différent. A chaque extrémité du canal se trouve un programme, l'agent MCA. A l'une des extrémités du canal, l'agent MCA demandeur extrait les messages de la file de transmission et les envoie sur le canal. A l'autre extrémité du canal, l'agent MCA répondeur reçoit les messages et les délivre au gestionnaire de files d'attente éloignées.

Un agent MCA demandeur peut être associé à un canal émetteur, serveur ou demandeur. Un agent MCA répondeur peut être associé à n'importe quel type de canal de transmission de messages.

IBM MQ prend en charge les combinaisons suivantes de types de canaux aux deux extrémités d'une connexion :

Canal appelant		Direction du flux de messages	Canal répondeur	
Type de canal	Programme d'écoute requis ?		Programme d'écoute requis ?	Type de canal
Emetteur	Non	Appelant à répondeur	Oui	Récepteur
serveur	Non	Appelant à répondeur	Oui	Récepteur
serveur	Non	Appelant à répondeur	Oui	Demandeur
Demandeur	Non	Répondeur à appelant	Oui	serveur

<b>Canal appelant</b>		<b>Direction du flux de messages</b>	<b>Canal répondeur</b>	
<b>Type de canal</b>	<b>Programme d'écoute requis ?</b>		<b>Programme d'écoute requis ?</b>	<b>Type de canal</b>
Demandeur	Oui	Répondeur à appelant	Oui	Emetteur

## Canaux MQI

Les canaux MQI peuvent être de l'un des types suivants :

Type de canal MQI	Description
Connexion serveur	Un canal de connexion serveur est un canal MQI bidirectionnel qui est utilisé pour connecter un client IBM MQ à un serveur IBM MQ. Le canal de connexion serveur constitue l'extrémité serveur du canal.
Connexion client	Un canal de connexion client est un canal MQI bidirectionnel qui est utilisé pour connecter un client IBM MQ à un serveur IBM MQ. IBM MQ Explorer utilise également des connexions client pour se connecter aux gestionnaires de files d'attente éloignées. Le canal de connexion client constitue l'extrémité client du canal. Lorsque vous créez un canal de connexion client, un fichier est créé sur l'ordinateur hébergeant le gestionnaire de files d'attente. Vous devez ensuite copier ce fichier de connexion client sur l'ordinateur client IBM MQ.

## Communications

Les IBM MQ MQI clients utilisent des canaux MQI pour communiquer avec le serveur.

Une définition de canal doit être créée sur les deux extrémités IBM MQ MQI client et serveur de la connexion. La procédure de création de définitions de canal est expliquée dans la rubrique [Définition de canaux MQI](#).

Les protocoles de transmission possibles sont présentés dans le tableau suivant :

*Tableau 1. Protocoles de transmission pour les canaux MQI*

Plateforme client	LU 6.2	TCP/IP	NetBIOS	SPX
 IBM i		Oui		
  Systèmes AIX and Linux	Oui <sup>1</sup>	Oui		
 Windows	Oui	Oui	Oui	Oui

### Remarque :

-  LU6.2 n'est pas pris en charge sur les plateformes suivantes :
  - Linux (plateforme POWER)
  - Linux (plateforme x86-64)
  - Linux (plateforme zSeries s390x)

Protocoles de transmission - Combinaison de plateformes IBM MQ MQI client et serveur indique les combinaisons possibles de plateformes IBM MQ MQI client et serveur en utilisant ces protocoles de transmission.

Une application IBM MQ sur un IBM MQ MQI client peut utiliser tous les appels MQI de la même façon que lorsque le gestionnaire de files d'attente est local. **MQCONN** ou **MQCONNX** associe l'application IBM MQ au gestionnaire de files d'attente sélectionné, afin de créer un *descripteur de connexion*. Les autres appels utilisant ce descripteur de connexion sont ensuite traités par le gestionnaire de files d'attente connecté. La communication du IBM MQ MQI client nécessite une connexion active entre le client et le serveur, par opposition à la communication entre les gestionnaires de files d'attente, qui est indépendante de la connexion et sans contrainte de temps.

Le protocole de communication est indiqué à l'aide de la définition de canal et n'a pas d'incidence sur l'application. Par exemple, une application Windows peut se connecter à un gestionnaire de files d'attente sur TCP/IP et à un autre gestionnaire de files d'attente sur NetBIOS.

## Considérations relative aux performances

Le protocole de communication que vous utilisez peut avoir une incidence sur les performances du système client et serveur IBM MQ. Pour une liaison commutée sur une ligne téléphonique lente, il convient d'utiliser la compression de canal IBM MQ.

## Canaux de connexion client

Les *canaux de connexion client* sont des objets fournissant un chemin de communication d'un IBM MQ MQI client vers un gestionnaire de files d'attente.

Les canaux de connexion client sont utilisés dans le cadre de la mise en file d'attente répartie pour déplacer des messages entre un gestionnaire de files d'attente et un client. Ils protègent les applications des protocoles de communication sous-jacents. Le client peut résider sur la même plateforme que le gestionnaire de files d'attente ou sur une plateforme différente.

### Tâches associées

Configuration des connexions entre le serveur et le client

## Classes d'archivage

Une classe d'archivage établit la correspondance (mappe) une ou plusieurs files d'attente avec un ensemble de pages.

Cela signifie que les messages destinés à cette file d'attente sont stockés (sous réserve d'une mise en mémoire tampon) sur cet ensemble de pages.

Les classes d'archivage ne sont prises en charge que sur IBM MQ for z/OS.

Pour plus d'informations sur les classes d'archivage, voir [Planification sur z/OS](#).

## Programmes d'écoute

Les *programmes d'écoute* sont des processus qui acceptent les demandes de réseau d'autres gestionnaires de files d'attente ou d'applications client et qui démarrent des canaux associés.

Les *processus d'écoute* peuvent être lancés à l'aide de la commande de contrôle **runmqtsr**.

Les *objets programme d'écoute* sont des objets IBM MQ qui permettent de gérer le démarrage et l'arrêt des processus d'écoute dans le cadre d'un gestionnaire de files d'attente. En définissant des attributs d'un objet programme d'écoute :

- Vous configurez le processus d'écoute.
- Vous indiquez si le processus d'écoute démarre et s'arrête automatiquement lors du démarrage et de l'arrêt du gestionnaire de files d'attente.

**Important :**  Les objets de programme d'écoute ne sont pas pris en charge sur IBM MQ for z/OS. Pour plus d'informations sur l'implémentation de l'écoute par IBM MQ for z/OS en utilisant l'initiateur de canal, voir «L'initiateur de canal sous z/OS», à la page 171.

### Référence associée

[runmq1sr](#) (exécution du programme d'écoute)

## Services

Les objets *service* permettent de définir les programmes à exécuter lors du démarrage ou de l'arrêt d'un gestionnaire de files d'attente.

Les programmes doivent être de types :

### Serveurs

Un *serveur* est un objet service dont le paramètre SERVTYPE est désigné par SERVER. Un objet service serveur est la définition d'un programme qui sera exécuté lors du démarrage d'un gestionnaire de files d'attente indiqué. Une seule instance d'un processus serveur peut être exécutée à un moment donné. Lors de l'exécution, le statut d'un processus serveur peut être surveillé à l'aide de la commande MQSC, DISPLAY SVSTATUS. En général, les objets de service serveur sont des définitions de programme, telles que les gestionnaires de messages non livrés ou les moniteurs de déclenchement ; cependant, les programmes pouvant être exécutés ne sont pas limités à ceux fournis avec IBM MQ. En outre, un objet service serveur peut être défini pour inclure une commande à exécuter lorsque le gestionnaire de files indiqué est arrêté pour mettre fin au programme.

### Commandes

Une *commande* est un objet service dont le paramètre SERVTYPE est désigné par COMMAND. Un objet service de commande est la définition d'un programme qui sera exécuté lors du démarrage ou de l'arrêt d'un gestionnaire de files d'attente indiqué. Plusieurs instances d'un processus de commande peuvent être exécutées en même temps. Les objets service de commande diffèrent des objets service serveur dans la mesure où, une fois le programme exécuté, le gestionnaire de files d'attente ne surveillera pas le programme. En général, les objets service de commande sont des définitions de programmes éphémères et exécutent une tâche spécifique telle que le démarrage d'une ou de plusieurs autres tâches.

**Important :**  Les objets de service ne sont pas pris en charge sous IBM MQ for z/OS.

### Concepts associés

[Utilisation des services](#)

## Objets de rubrique

Un *objet rubrique* est un objet IBM MQ qui permet d'affecter des attributs spécifiques différents des attributs par défaut à des rubriques.

Une *rubrique* est définie par une application qui publie une *chaîne de rubrique* ou *s'y abonne*. Une chaîne de rubrique peut spécifier une hiérarchie de rubriques en les séparant par une barre oblique (/). Cela peut être visualisé par une *arborescence de rubriques*. Par exemple, si une application publie dans les chaînes de rubrique /Sport/American Football et /Sport/Soccer, une arborescence de rubriques est créée avec un noeud parent Sport avec deux enfants, American Football et Soccer.

Les rubriques héritent leurs attributs du premier noeud d'administration parent trouvé dans leur arborescence de rubriques. Si une arborescence de rubriques particulière ne contient aucun noeud de rubrique d'administration, toutes les rubriques héritent leurs attributs de l'objet de rubrique de base, SYSTEM.BASE.TOPIC.

Vous pouvez créer un objet de rubrique sur n'importe quel noeud d'une arborescence de rubriques en définissant la chaîne de rubrique du noeud dans l'attribut TOPICSTR de l'objet de rubrique. Vous pouvez également définir d'autres attributs pour le noeud de rubrique d'administration. Pour plus d'informations sur ces attributs, voir [Les commandes MQSC](#) ou [Administration de l'automatisation à l'aide de commandes PCF](#). Par défaut, chaque objet de rubrique hérite des attributs de son noeud de rubrique administratif parent le plus proche.

Les objets de rubrique peuvent être également utilisés pour masquer aux développeurs d'applications la totalité de l'arborescence des rubriques. Si un objet de rubrique nommé FOOTBALL . US est créé pour la rubrique /Sport/American Football, une application peut publier ou s'abonner à l'objet nommé FOOTBALL . US au lieu de la chaîne /Sport/American Football avec le même résultat.

Si vous entrez un caractère #, +, / ou \* dans une chaîne de rubrique dans un objet de rubrique, le caractère est créé comme caractère normal dans la chaîne, car il est considéré faire partie de la chaîne de rubrique associée à l'objet de rubrique.

Pour plus d'informations sur les objets de rubrique, voir [«Messagerie de type publication/abonnement»](#), à la page 66.

## Attribution de nom aux objets IBM MQ

La convention d'attribution de nom adoptée pour les objets IBM MQ dépend de l'objet. Le nom des machines et les ID utilisateur employés avec IBM MQ sont également soumis à certaines restrictions de dénomination.

Chaque instance d'un gestionnaire de files d'attente est désignée par son nom. Ce nom doit être unique au sein du réseau des gestionnaires de files d'attente interconnectés, de sorte qu'un gestionnaire de files d'attente puisse identifier clairement le gestionnaire de files d'attente cible auquel un message donné est envoyé.

Pour les autres types d'objet, un nom est associé à chaque objet qui peut être désigné par ce nom. Ces noms doivent être uniques au sein d'un gestionnaire de files d'attente et d'un type d'objet. Par exemple, vous pouvez avoir une file d'attente et un processus du même nom, mais vous ne pouvez pas avoir deux files d'attente du même nom.

Dans IBM MQ, les noms peuvent comporter au maximum 48 caractères, à l'exception des *canaux* qui peuvent comporter au maximum 20 caractères. Pour plus d'informations sur l'attribution de nom aux objets IBM MQ, voir [«Règles d'appellation des objets IBM MQ»](#), à la page 38.

Le nom des machines et les ID utilisateur employés avec IBM MQ sont également soumis à certaines restrictions de dénomination :

- Vérifiez que le nom du poste ne contient aucun espace. En effet, IBM MQ ne prend pas en charge les noms de postes comportant des espaces. Si toutefois vous installez IBM MQ sur un tel poste, vous ne pourrez pas créer de gestionnaire de files d'attente.
- En ce qui concerne les droits IBM MQ, les ID utilisateur et les noms de groupes ne doivent pas comporter plus de vingt caractères (sans espaces).
-  Un serveur IBM MQ for Windows ne prend pas en charge la connexion d'un IBM MQ MQI client si le client est exécuté sous un ID utilisateur qui contient le caractère @, par exemple, abc@d.

### Concepts associés

[«Noms de fichier IBM MQ»](#), à la page 41

Chaque objet gestionnaire de files d'attente, file d'attente, définition de processus, liste de noms, canal, canal de connexion client, programme d'écoute, service et informations d'authentification IBM MQ est représenté par un fichier. Dans la mesure où les noms d'objet ne sont pas nécessairement des noms de fichier valides, le gestionnaire de files d'attente convertit, si nécessaire, le nom d'objet en nom de fichier valide.

### Référence associée

[«Règles d'appellation des objets IBM MQ»](#), à la page 38

Les noms d'objet IBM MQ ont des longueurs maximales et tiennent compte de la casse. Les caractères ne sont pas tous pris en charge pour chaque type d'objet et la plupart des objets ont des règles concernant l'unicité des noms.

## Règles d'appellation des objets IBM MQ

Les noms d'objet IBM MQ ont des longueurs maximales et tiennent compte de la casse. Les caractères ne sont pas tous pris en charge pour chaque type d'objet et la plupart des objets ont des règles concernant l'unicité des noms.

Il existe un grand nombre de types d'objets IBM MQ différents et les objets d'un certain type peuvent avoir tous le même nom, car ils existent dans des espaces nom d'objet différents. Par exemple, une file d'attente locale et un canal émetteur peuvent porter le même nom. Toutefois, un objet ne peut pas porter le même nom qu'un autre objet dans un même espace nom. Par exemple, une file d'attente locale ne peut pas porter le nom d'une file d'attente modèle et un canal émetteur ne peut pas porter le nom d'un canal récepteur.

Les objets IBM MQ suivants existent dans des espaces nom d'objet distincts :

- Informations d'authentification
- Canal
- Canal client
- Programme d'écoute
- Liste de noms
- Processus
- File d'attente
- Service
- Classe d'archivage
- Abonnement
- Topic

### Nombre de caractères des noms d'objet

En règle générale, les noms d'objet IBM MQ peuvent contenir jusqu'à 48 caractères. Cette règle s'applique aux objets suivants :

- Informations d'authentification
- Cluster
- Programme d'écoute
- Liste de noms
- Définition de processus
- File d'attente
- Gestionnaire de files d'attente
- Service
- Abonnement
- Topic

Il existe des restrictions :

1.  Sur les systèmes z/OS, les gestionnaires de files d'attente doivent comporter 4 caractères maximum et doivent être en majuscules et numériques.
2. Les noms d'objet de canal et de canal de connexion client peuvent comporter jusqu'à 20 caractères. Pour plus d'informations sur les canaux, voir [Définition des canaux](#).
3. Les chaînes de rubrique peuvent avoir une longueur de 10 240 caractères maximum. Tous les noms d'objet IBM MQ tiennent compte de la casse.
4. Les noms d'abonnement peuvent avoir une longueur de 10 240 caractères maximum et contenir des espaces.

5. La longueur maximale des noms de classe de stockage est de 8 caractères.
6. La longueur maximale des noms de structure d'unité de couplage est de 12 caractères.

## Caractères dans les noms d'objet

Caractères valides pour les noms d'objet IBM MQ :

caractères	Restrictions
Majuscules A - Z	<ul style="list-style-type: none"> <li>• Aucun</li> </ul>
Minuscules a - z	<ul style="list-style-type: none"> <li>• Dans les scripts MQSC, les noms en minuscules doivent être placés entre des guillemets simples. Cela empêche la conversion des caractères minuscules en majuscules.</li> <li>• Les systèmes qui utilisent EBCDIC Katakana ne peuvent pas utiliser les minuscules a-z dans les noms d'objet.</li> <li>•  Il peut exister des restrictions lorsque vous utilisez des minuscules sur les systèmes z/OS. Par exemple, les noms de gestionnaire de files d'attente ne peuvent pas contenir des minuscules.</li> <li>•  Sur les systèmes IBM i, lorsque des commandes CL sont utilisées, les noms en minuscules doivent être placés entre des guillemets simples. Cela empêche la conversion des caractères minuscules en majuscules.</li> </ul>
Caractères numériques 0 - 9	<ul style="list-style-type: none"> <li>• Aucun</li> </ul>
le point (.)	<ul style="list-style-type: none"> <li>• Aucun</li> </ul>
le caractère de soulignement (_)	<ul style="list-style-type: none"> <li>•  Aucun</li> <li>•  Evitez d'utiliser des noms avec des caractères de soulignement de début et de fin, car les opérations et les panneaux de configuration IBM MQ for z/OS ne peuvent pas les traiter.</li> </ul>
la barre oblique (/)	<ul style="list-style-type: none"> <li>•  Sur les systèmes Windows, le premier caractère d'un nom de gestionnaire de files d'attente ne peut pas être une barre oblique.</li> <li>•  Sur les systèmes IBM i, lorsque des commandes CL sont utilisées, les noms contenant une barre oblique doivent être placés entre des guillemets simples.</li> <li>•  Aucun</li> </ul>

caractères	Restrictions
le symbole de pourcentage (%)	<ul style="list-style-type: none"> <li data-bbox="820 199 1079 241">•  ALW Aucun</li> <li data-bbox="820 262 1461 430">•  z/OS Si vous utilisez RACF comme gestionnaire de sécurité pour IBM MQ for z/OS, n'utilisez pas % dans les noms d'objet, car les noms ne sont pas inclus dans les vérifications de sécurité lorsque des profils génériques RACF sont utilisés.</li> <li data-bbox="820 451 1461 577">•  IBM i Sur les systèmes IBM i, lorsque des commandes CL sont utilisées, les noms contenant le symbole de pourcentage doivent être placés entre des guillemets simples.</li> </ul>

Il existe des règles générales concernant les caractères dans les noms d'objet :

1. Les espaces de début ou imbriqués ne sont pas autorisés.
2. Les caractères en langue nationale ne sont pas autorisés.
3. Les noms dont la longueur est inférieure à la longueur totale de la zone peuvent être complétés sur la droite avec des espaces. Tous les noms courts retournés par le gestionnaire de files d'attente sont toujours complétés à droite avec des espaces.

## Noms des files d'attente

Un nom de file d'attente a deux parties :

- Nom d'un gestionnaire de files d'attente
- Nom local de la file d'attente que connaît le gestionnaire de files d'attente

Chaque partie du nom de file d'attente comporte 48 caractères.

Pour faire référence à une file d'attente locale, vous pouvez omettre le nom du gestionnaire de files d'attente (en le remplaçant par des espaces ou en utilisant un caractère null de début). Toutefois, tous les noms de file d'attente retournés à un programme par IBM MQ contiennent le nom du gestionnaire de files d'attente.

 Une file d'attente partagée, accessible à n'importe quel gestionnaire de files d'attente dans son groupe de partage de files d'attente, ne peut pas porter le nom d'une file d'attente local non partagée dans le même groupe de partage de files d'attente. Cette restriction empêche une application d'ouvrir par erreur une file d'attente partagée en pensant ouvrir une file d'attente locale ou inversement. Les files d'attente partagées et les groupes de partage de files d'attente sont disponibles uniquement dans IBM MQ for z/OS.

Pour faire référence à une file d'attente éloignée, un programme doit inclure le nom du gestionnaire de files d'attente dans le nom de file d'attente complet ou il doit exister une définition locale de la file d'attente éloignée.

Lorsqu'une application utilise un nom de file d'attente, ce nom peut être un nom de file d'attente locale (ou un alias d'une telle file d'attente) ou le nom d'une définition locale d'une file d'attente éloignée, mais l'application n'a pas besoin de les distinguer, sauf si elle doit obtenir un message de la file d'attente (lorsque la file d'attente doit être locale). Lorsque l'application ouvre l'objet file d'attente, l'appel MQOPEN exécute une résolution de nom pour déterminer la file d'attente dans laquelle elle doit exécuter les opérations suivantes. Cela implique que l'application n'a pas de dépendance intégrée dans les files d'attente définies dans des emplacements donnés dans un réseau de gestionnaires de files d'attente. Par conséquent, si un administrateur système déplace les files d'attente dans un réseau et modifie leur définition, les applications qui utilisent ces files d'attente n'ont pas besoin d'être modifiées.

## Noms d'objet réservés

Les noms d'objet commençant par SYSTEM . sont réservés aux objets définis par le gestionnaire de files d'attente. Vous pouvez utiliser les commandes **Alter**, **Define** et **Replace** pour changer ces définitions d'objet en fonction de l'installation. Les noms qui sont définis pour IBM MQ sont affichés en entier dans [Noms de file d'attente](#).

 Sur IBM MQ for z/OS, le nom de la structure d'application de l'unité de couplage CSQSYSAPPL est réservé.

### Concepts associés

[Nom d'installation sous AIX, Linux, and Windows](#)

## Noms de fichier IBM MQ

Chaque objet gestionnaire de files d'attente, file d'attente, définition de processus, liste de noms, canal, canal de connexion client, programme d'écoute, service et informations d'authentification IBM MQ est représenté par un fichier. Dans la mesure où les noms d'objet ne sont pas nécessairement des noms de fichier valides, le gestionnaire de files d'attente convertit, si nécessaire, le nom d'objet en nom de fichier valide.

Le chemin d'accès par défaut à un répertoire de gestionnaires de files d'attente est le suivant :

- Un préfixe, défini dans les informations de configuration IBM MQ :
  -   Sous AIX and Linux, le préfixe par défaut est /var/mqm. Celui-ci est configuré dans la section DefaultPrefix du fichier de configuration mqs.ini.
  -  Sur les systèmes Windows 32 bits, le préfixe par défaut est C:\Program Files (x86)\IBM\WebSphere MQ. Sur les systèmes Windows 64 bits, le préfixe par défaut est C:\Program Files\IBM\MQ. Pour les installations 32 bits et 64 bits, les répertoires de données sont installés dans C:\ProgramData\IBM\MQ. Celui-ci est configuré dans la section DefaultPrefix du fichier de configuration mqs.ini.

Le cas échéant, le préfixe peut être modifié depuis la page de propriétés IBM MQ de l'explorateur IBM MQ, sinon, éditez le fichier de configuration mqs . ini manuellement.

- Le nom du gestionnaire de files d'attente est converti en nom de répertoire valide. Par exemple, le gestionnaire de files d'attente :

```
queue.manager
```

est représenté sous forme de :

```
queue!manager
```

Ce processus est appelé *transformation de noms*.

Dans IBM MQ, vous pouvez attribuer aux gestionnaires de files d'attente un nom contenant jusqu'à 48 caractères.

Par exemple, vous pouvez attribuer à un gestionnaire de files d'attente le nom suivant :

```
QUEUE.MANAGER.ACCOUNTING.SERVICES
```

Cependant, chaque gestionnaire de files d'attente est représenté par un fichier et la longueur maximale d'un nom de fichier est limitée, ainsi que le nombre de caractères pouvant être utilisés dans le nom. En conséquence, les noms de fichiers représentant des objets sont automatiquement convertis en fonction des exigences du système de fichiers.

Les règles régissant la transformation d'un nom de gestionnaire de files d'attente sont les suivantes :

## 1. Transformation de caractères individuels :

- Depuis . à !
- Depuis / vers &

## 2. Si le nom n'est toujours pas valide :

- a. Tronquez-le à huit caractères
- b. Ajoutez un suffixe numérique de trois caractères

Par exemple, en utilisant le préfixe par défaut et un gestionnaire de files d'attente intitulé `queue.manager` :

- **Windows** Sous Windows doté du système NTFS ou FAT32, le nom du gestionnaire de files d'attente devient :

```
C:\Program Files\IBM\MQ\mqgrs\queue!manager
```

- **Windows** Sous Windows doté du système FAT, le nom du gestionnaire de files d'attente devient :

```
C:\Program Files\IBM\MQ\mqgrs\queue!ma
```

- **Linux** **AIX** Sous AIX and Linux, le nom du gestionnaire de files d'attente devient :

```
/var/mqm/mqgrs/queue!manager
```

L'algorithme de transformation fait également la distinction entre les noms dont seule la casse diffère sur les systèmes de fichiers qui ne sont pas sensibles à la casse.

## Transformation de nom d'objet

Les noms d'objet ne sont pas nécessairement des noms de système de fichiers valides. Vous devrez peut-être convertir les noms d'objet. La méthode utilisée est différente de celle des noms de gestionnaire de files d'attente car, bien qu'il n'existe que quelques noms de gestionnaire de files d'attente sur chaque machine, il peut y avoir un grand nombre d'autres objets pour chaque gestionnaire de files d'attente. Les objets files d'attente, définitions de processus, listes de noms, canaux, canaux de connexion client, programmes d'écoute et informations d'authentification sont représentés dans le système de fichiers.

Lorsqu'un nouveau nom est généré par le processus de transformation, il n'y a aucune relation simple avec le nom d'objet d'origine. Vous pouvez utiliser la commande **dspmqls** pour la transformation entre les noms d'objet réels et convertis.

### Référence associée

**dspmqls** (affichage des noms de fichier)

## IBM i Noms d'objet sur IBM i

Un gestionnaire de files d'attente est associé à une bibliothèque de gestionnaires de files d'attente qui possède un nom unique. Il se peut que les noms de gestionnaire de files d'attente et les noms d'objet doivent être transformés pour répondre aux exigences du système de fichiers intégré (IFS) d'IBM i.

Lorsqu'un gestionnaire de files d'attente est créé, IBM MQ lui associe une bibliothèque de gestionnaire de files d'attente. Un nom unique est attribué à cette bibliothèque de gestionnaire de files d'attente ; il comporte au maximum 10 caractères et est basé en grande partie sur le nom de gestionnaire de files d'attente défini par l'utilisateur. Le gestionnaire de files d'attente et la bibliothèque de gestionnaire de files d'attente sont placés dans un répertoire qui est également basé sur le nom de gestionnaire de files d'attente à l'aide du préfixe `/QIBM/UserData/mqm`. Voici un exemple de gestionnaire de files d'attente, de bibliothèque de gestionnaire de files d'attente et de répertoire :

Nom gest. de files	ORANGE
Nom de bibliothèque du gestionnaire de files d'attente	QMORANGE
Répertoire	/QIBM/UserData/mqm/ORANGE

Tous les noms de gestionnaire de files d'attente et bibliothèques de gestionnaire de files d'attente sont écrits dans les strophes du fichier `/QIBM/UserData/mqm/mqs.ini`.

## Répertoires et fichiers IFS d'IBM MQ

IFS (IBM i Integrated File System) est largement utilisé par IBM MQ pour stocker des données. Pour plus d'informations sur le système IFS, consultez le document *Integrated File System Introduction*.

Chaque objet IBM MQ, tel qu'un canal ou une file d'attente, est représenté par un fichier. Dans la mesure où les noms d'objet ne sont pas nécessairement des noms de fichier valides, le gestionnaire de files d'attente convertit, si nécessaire, le nom d'objet en nom de fichier valide.

Le chemin d'accès à un répertoire de gestionnaires de files d'attente est formé à partir des éléments suivants :

- Un préfixe, qui est défini dans le fichier de configuration du gestionnaire de files d'attente, `qm.ini`. Le préfixe par défaut est `/QIBM/UserData/mqm`.
- Un littéral, `qmgrs`.
- Un nom du gestionnaire de files d'attente codé, qui correspond au nom de gestionnaire de files d'attente converti en nom de répertoire valide. Par exemple, le gestionnaire de files d'attente `queue/manager` est représenté par `queue&manager`.

Ce processus est désigné par transformation de noms.

## Transformation de noms de gestionnaire de files d'attente IFS

Dans IBM MQ, vous pouvez attribuer aux gestionnaires de files d'attente un nom contenant jusqu'à 48 caractères.

Par exemple, vous pouvez attribuer à un gestionnaire de files d'attente le nom `QUEUE/MANAGER/ACCOUNTING/SERVICES`. A l'instar d'une bibliothèque créée pour chaque gestionnaire de files d'attente, chaque gestionnaire de files d'attente est également représenté par un fichier. En raison de points de code différents dans EBCDIC, le nombre de caractères pouvant être utilisés dans le nom est limité. En conséquence, les noms de fichiers IFS représentant des objets sont automatiquement convertis en fonction des exigences du système de fichiers.

À partir de l'exemple d'un gestionnaire de files d'attente portant le nom `queue/manager`, transformant le caractère `/` en `&` et en supposant le préfixe par défaut, le nom du gestionnaire de files d'attente dans IBM MQ for IBM i devient `/QIBM/UserData/mqm/qmgrs/queue&manager`.

## Transformation de nom d'objet

Les noms d'objet ne sont pas nécessairement des noms de système de fichiers valides ; par conséquent, il est peut-être nécessaire de convertir les noms d'objet. La méthode utilisée est différente de celle des noms de gestionnaire de files d'attente car, bien qu'il n'existe que quelques noms de gestionnaire de files d'attente pour chaque machine, il peut y avoir un grand nombre d'autres objets pour chaque gestionnaire de files d'attente. Seules les définitions de processus, les files d'attente et les listes de noms sont représentées dans le système de fichiers ; ces critères n'ont pas d'incidence sur les canaux.

Lorsqu'un nouveau nom est généré par le processus de transformation, il n'y a aucune relation simple avec le nom d'objet d'origine. Vous pouvez utiliser la commande `DSPMQMOBJN` pour afficher les noms convertis des objets IBM MQ.

## Attributs d'objet

Les propriétés d'un objet sont définies par ses attributs. Certains peuvent être spécifiés et d'autres ne peuvent être que consultés.

Par exemple, la longueur de message maximale pouvant être prise en charge par une file d'attente est définie par son attribut **MaxMsgLength** ; vous pouvez spécifier cet attribut lorsque vous créez une file d'attente. L'attribut **DefinitionType** indique comment la file d'attente a été créée ; vous pouvez uniquement afficher cet attribut.

Dans IBM MQ, deux méthodes permettent de faire référence à un attribut :

- Utilisation de son nom PCF, par exemple, **MaxMsgLength**.
- Utilisation de son nom de commande MQSC, par exemple, MAXMSGL.

## Groupes de partage de files d'attente

Les gestionnaires de files d'attente pouvant accéder au même ensemble de files d'attente partagées forment un groupe appelé *groupe de partage de files d'attente* et peuvent communiquer entre eux à l'aide d'une unité de couplage qui stocke les files d'attente partagées.

Une file d'attente partagée est un type de file d'attente locale comportant des messages accessibles à un ou plusieurs gestionnaires de files d'attente d'un groupe de partage de files d'attente. Il ne s'agit pas d'une file d'attente partagée par plusieurs applications, à l'aide du même gestionnaire de files d'attente.

Le nom des groupes de partage de files d'attente comporte jusqu'à quatre caractères. Il doit être unique sur votre réseau et être différent de celui d'un gestionnaire de files d'attente.

Les groupes de partage de files d'attente ne sont pas des objets proprement dits, mais sont mentionnés ici par souci de commodité.

**Important :** les files d'attente partagées et les groupes de partage de files d'attente sont pris en charge dans IBM MQ for z/OS uniquement.

### Concepts associés

«Files d'attente partagées et groupes de partage de files d'attente», à la page 176

Vous avez la possibilité d'utiliser des files d'attentes partagées et des groupes de partage de files d'attentes pour implémenter la haute disponibilité des ressources IBM MQ. Les files d'attente partagées et les groupes de partage de files d'attente sont des fonctions propres à IBM MQ for z/OS sur la plateforme z/OS.

## Objets par défaut du système

Les *objets par défaut du système* sont un ensemble de définitions d'objet créées automatiquement chaque fois qu'un gestionnaire de files d'attente est créé.

Vous pouvez copier et modifier l'une quelconque de ces définitions d'objet à des fins d'utilisation dans les applications de votre installation.

Les noms d'objet par défaut ont la racine SYSTEM ; par exemple, la file d'attente locale par défaut est SYSTEM.DEFAULT.LOCAL.QUEUE et le canal récepteur par défaut est SYSTEM.DEF.RECEIVER. Vous ne pouvez pas renommer ces objets ; des objets par défaut de ces noms sont requis.

Lors de la définition d'un objet, tous les attributs non spécifiés explicitement sont copiés à partir de l'objet par défaut approprié. Par exemple, si vous définissez une file d'attente locale, les attributs que vous ne spécifiez pas sont extraits de la file d'attente par défaut SYSTEM.DEFAULT.LOCAL.QUEUE.

### Référence associée

Objets système et par défaut

## Mise en file d'attente répartie et clusters

La mise en file d'attente répartie permet d'envoyer des messages d'un gestionnaire de files d'attente à un autre. Le gestionnaire de files d'attente récepteur peut résider sur la même machine ou sur une autre, à proximité ou à l'autre bout du monde. Il peut s'exécuter sur la même plateforme que le gestionnaire de files d'attente local ou se trouver sur n'importe laquelle des plateformes prises en charge par IBM MQ. Vous pouvez définir manuellement toutes les connexions dans un environnement de mise en file d'attente répartie, ou créer un cluster et laisser IBM MQ définir la plupart des informations de connexion automatiquement.

### Files d'attente réparties

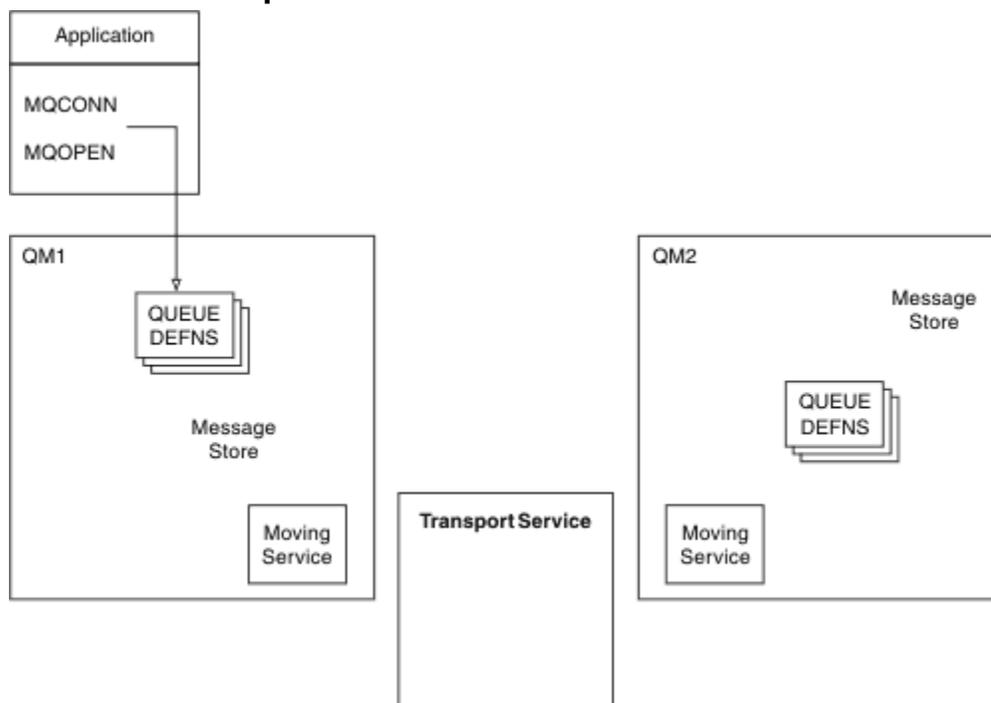


Figure 4. Présentation des composants de la mise en file d'attente répartie

Dans la figure précédente :

- Une application utilise l'appel MQCONN pour se connecter à un gestionnaire de files d'attente. L'application utilise ensuite l'appel MQOPEN pour ouvrir une file d'attente de sorte qu'elle puisse y insérer des messages.
- Chaque gestionnaire de files d'attente comporte une définition pour chacune de ses files d'attente. Il peut avoir des définitions de *files d'attente locales* (c'est-à-dire hébergées par ce gestionnaire de files d'attente) et des définitions de *files d'attente éloignées* (c'est-à-dire, hébergées par d'autres gestionnaires de files d'attente).
- Si les messages sont destinés à une file d'attente éloignée, le gestionnaire de files d'attente local les conserve dans une *file d'attente de transmission*, qui les maintient dans un emplacement de stockage des messages jusqu'à ce qu'ils puissent être transmis au gestionnaire de files d'attente éloignées.
- Chaque gestionnaire de files d'attente contient un logiciel de communication appelé *service de déplacement*, utilisé par le gestionnaire de files d'attente pour communiquer avec d'autres gestionnaires de files d'attente.
- Le *service de transfert* est indépendant du gestionnaire de files d'attente et peut correspondre à l'un des suivants (en fonction de la plateforme) :
  - Communication avancée de programme à programme SNA (Systems Network Architecture)
  - protocole TCP/IP

- Système NetBIOS (Network Basic Input/Output System)
- SPX (Sequenced Packet Exchange)

### Composants requis pour l'envoi d'un message

Si un message doit être envoyé à un gestionnaire de files d'attente éloignées, le gestionnaire de files d'attente local a besoin des définitions pour une *file d'attente de transmission* et un *canal*. Un canal est une liaison de communication unidirectionnelle entre deux gestionnaires de files d'attente. Il peut transporter des messages destinés à n'importe quel nombre de files d'attente sur le gestionnaire de files d'attente éloignées.

Chaque extrémité d'un canal comporte une définition distincte qui la définit, par exemple, comme extrémité émettrice ou extrémité réceptrice. Un canal simple est constitué d'une définition de canal *émetteur* sur le gestionnaire de files d'attente local et d'une définition de canal *récepteur* sur le gestionnaire de files d'attente éloignées. Ces deux définitions doivent porter le même nom et former conjointement un canal.

Le logiciel traitant l'envoi et la réception des messages est appelé *agent MCA* (Message Channel Agent). Un *Agent MCA* se trouve à chaque extrémité d'un canal.

Chaque gestionnaire de files d'attente doit posséder une *file d'attente de rebut* (également appelée *file d'attente des messages non livrés*). Les messages sont insérés dans cette file d'attente s'ils ne peuvent pas être distribués à leur destination.

La figure suivante affiche la relation entre les gestionnaires de files d'attente, les files d'attente de transmission, les canaux et les agents MCA :

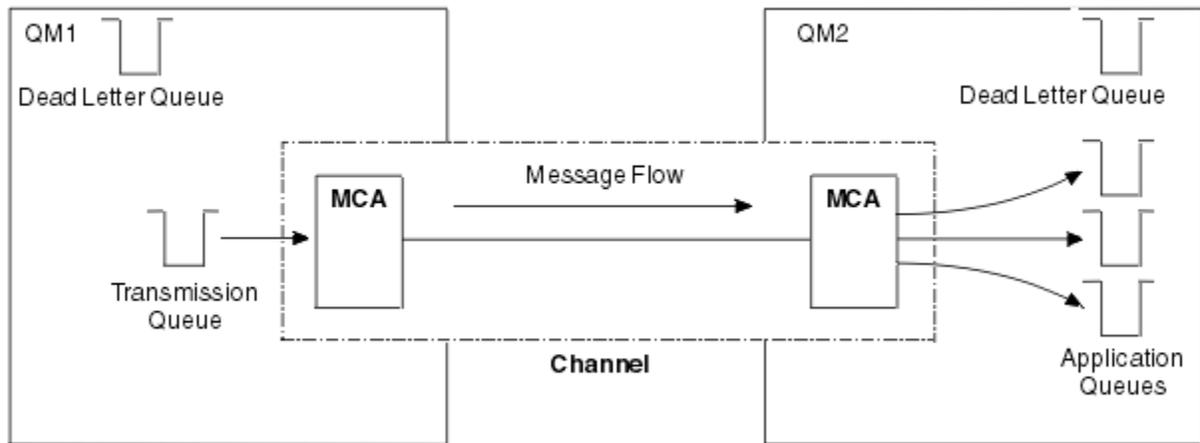


Figure 5. Envoi de messages

### Composants requis pour le renvoi d'un message

Si votre application requiert le renvoi des messages à partir du gestionnaire de files d'attente éloignées, vous devez définir un autre canal à exécuter dans le sens opposé entre les gestionnaires de files d'attente, comme présenté dans la figure suivante :

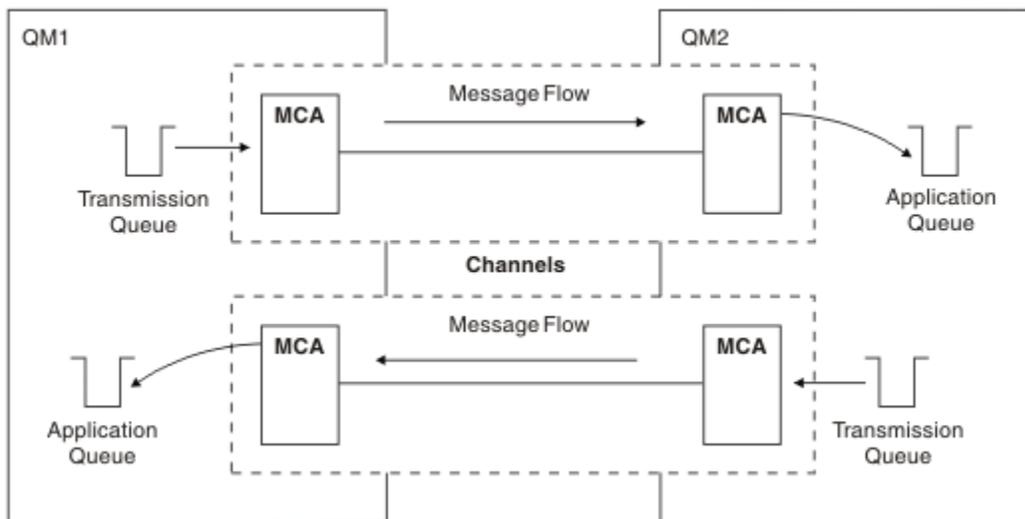


Figure 6. Envoi de messages dans les deux sens

## Groupes

Au lieu de définir manuellement toutes les connexions dans un environnement de mise en file d'attente répartie, vous pouvez regrouper un ensemble de gestionnaires de files d'attente dans un cluster. Lors de cette opération, les gestionnaires de files d'attente peuvent rendre les files d'attente qu'ils hébergent disponibles pour tous les autres gestionnaires de files d'attente du cluster sans avoir besoin de définitions de canal explicites, des définitions de file d'attente éloignée ou des files d'attente de transmission de chaque destination. Chaque gestionnaire de files d'attente d'un cluster dispose d'une file d'attente de transmission unique qui transmet des messages à tout autre gestionnaire de files d'attente du cluster. Pour chaque gestionnaire de files d'attente, il suffit de définir un seul canal récepteur de cluster et un seul canal émetteur de cluster ; tous les autres canaux sont automatiquement gérés par le cluster.

Un client IBM MQ peut se connecter à un gestionnaire de files d'attente membre d'un cluster, comme à n'importe quel autre gestionnaire de files d'attente. Comme dans le cas de la mise en file d'attente répartie configurée manuellement, vous utilisez l'appel MQPUT pour insérer un message dans une file d'attente de n'importe quel gestionnaire de files d'attente. Vous utilisez l'appel MQGET pour extraire des messages d'une file d'attente locale.

Les gestionnaires de files d'attente sur les plateformes qui prennent en charge les clusters n'ont pas besoin d'être membres d'un cluster. Vous pouvez également continuer de configurer manuellement la mise en file d'attente répartie et l'utiliser en plus ou à la place des clusters.

## Avantages de l'utilisation de clusters

La mise en cluster offre deux avantages essentiels :

- Les clusters simplifient l'administration de réseaux IBM MQ qui nécessitent généralement la configuration de nombreuses définitions d'objet pour les canaux, les files d'attente de transmission et les files d'attente éloignées. C'est en particulier le cas pour des réseaux de grande taille susceptibles d'être modifiés dans lesquels de nombreux gestionnaires de files d'attente doivent être interconnectés. Cette architecture est particulièrement difficile à configurer et gérer activement.
- Des clusters peuvent être utilisés pour répartir la charge de travail du trafic des messages entre les files d'attente et les gestionnaires de files d'attente d'un cluster. Vous pouvez ainsi répartir la charge de travail liée aux messages d'une seule file d'attente sur des instances équivalentes de celle-ci située sur plusieurs gestionnaires de files d'attente. Cette répartition de la charge de travail permet d'obtenir une résilience accrue lors de pannes système et d'améliorer les performances d'évolutivité des flux de messages particulièrement actifs dans un système. Dans un tel environnement, chacune des instances des files d'attente réparties dispose d'applications consommatrices qui traitent les messages. Pour plus d'informations, voir [Utilisation des clusters pour la gestion de la charge de travail](#).

## Procédure de routage des messages dans un cluster

Un cluster s'apparente à un réseau de gestionnaires de files d'attente géré par un administrateur système consciencieux. Lorsque vous définissez une file d'attente de cluster, l'administrateur système crée automatiquement des définitions de files d'attente éloignées de manière appropriée dans les autres gestionnaires de files d'attente.

Vous n'avez pas à créer de définitions de files d'attente de transmission, car IBM MQ fournit une file d'attente de transmission dans chaque gestionnaire de files d'attente du cluster. Cette file d'attente de transmission unique peut être utilisée pour transporter les messages vers n'importe quelle autre file d'attente du cluster. Vous n'êtes pas limité à l'utilisation d'une file d'attente de transmission unique. Un gestionnaire de files d'attente peut utiliser plusieurs files d'attente de transmission pour séparer les messages insérés dans chaque gestionnaire de files d'attente d'un cluster. En règle générale, un gestionnaire de files d'attente utilise une file d'attente de transmission de cluster unique. Vous pouvez modifier l'attribut de gestionnaire de files d'attente DEFCLXQ afin qu'un gestionnaire de files d'attente utilise une file d'attente de transmission de cluster différente pour chaque gestionnaire de files d'attente d'un cluster. Vous pouvez également définir les files d'attente de transmission de cluster manuellement.

Tous les gestionnaires de files d'attente qui rejoignent un cluster acceptent ce mode de fonctionnement. Ils envoient des informations sur eux-mêmes et sur les files d'attente qu'ils hébergent et ils reçoivent des informations sur les autres membres du cluster.

Pour vous assurer qu'aucune information n'est perdue lorsqu'un gestionnaire de files d'attente devient indisponible, vous désignez deux gestionnaires de files d'attente du cluster afin qu'ils agissent en tant que *référentiels complets*. Ces gestionnaires de files d'attente stockent un ensemble complet d'informations sur tous les gestionnaires de files d'attente et files d'attente du cluster. Les autres gestionnaires de files d'attente du cluster stockent uniquement les informations sur les gestionnaires de files d'attente et les files d'attente avec lesquels ils échangent des messages. Ces gestionnaires de files d'attente sont appelés *référentiels partiels*. Pour plus d'informations, voir [«Référentiel de cluster»](#), à la page 59.

Pour qu'un gestionnaire de files d'attente puisse appartenir à un cluster, le gestionnaire de files d'attente doit disposer de deux canaux, un canal émetteur de cluster et un canal récepteur de cluster :

- Un canal émetteur de cluster est un canal de communication, à l'instar d'un canal émetteur. Vous devez créer manuellement un canal émetteur de cluster dans un gestionnaire de files d'attente pour le connecter à un référentiel complet déjà membre du cluster.
- Un canal récepteur de cluster est un canal de communication, à l'instar d'un canal récepteur. Vous devez créer manuellement un canal récepteur de cluster. Le canal fait office de mécanisme pour que le gestionnaire de files d'attente puisse recevoir les communications du cluster.

Tous les autres canaux nécessaires pour les communications entre ce gestionnaire de files d'attente et d'autres membres du cluster sont créés automatiquement.

La figure suivante représente les composants d'un cluster nommé CLUSTER :

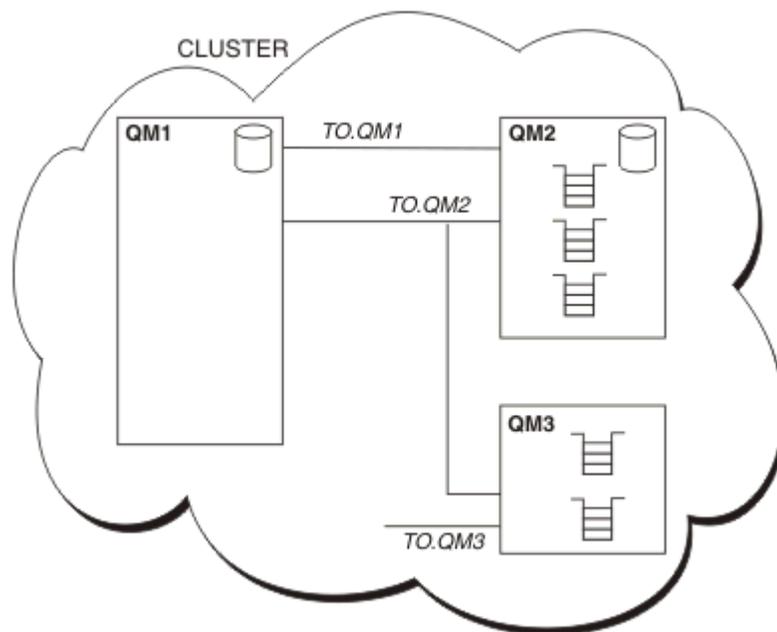


Figure 7. Cluster de gestionnaires de files d'attente

- CLUSTER contient trois gestionnaires de files d'attente : QM1, QM2 et QM3.
- QM1 et QM2 héberge des référentiels complets d'informations relatives aux gestionnaires de files d'attente et aux files d'attente du cluster.
- QM2 et QM3 héberge certaines files d'attente de cluster, c-à-d, des files d'attente accessibles à n'importe quel autre gestionnaire de files d'attente du cluster.
- Chaque gestionnaire de files d'attente comporte un canal récepteur de cluster appelé TO.qmgr sur lequel il peut recevoir des messages.
- Chaque gestionnaire de files d'attente comporte également un canal émetteur de cluster sur lequel il peut envoyer des informations à l'un des gestionnaires de files d'attente de référentiel.
- QM1 et QM3 envoient des informations au référentiel de QM2 et ce dernier envoie des informations au référentiel de QM1.

## Composants de la mise en file d'attente répartie

Les composants de la mise en file d'attente répartie sont les canaux de message, les agents MCA, les files d'attente de transmission, les initiateurs et programmes d'écoute de canal, et les programmes d'exit de canal. La définition de chaque extrémité d'un canal de transmission de messages peut être de différents types.

Les canaux de transmission de messages sont les canaux qui transfèrent des messages d'un gestionnaire de files d'attente vers un autre. Ne confondez pas les canaux de transmission de messages avec les canaux MQI. Il existe deux types de canal MQI : connexion serveur (SVRCONN) et connexion client (CLNTCONN). Pour plus d'informations, voir [Canaux](#).

La définition de chaque extrémité d'un canal de transmission de messages peut être de l'un des types suivants :

- Emetteur (SDR)
- Récepteur (RCVR)
- Serveur (SVR)
- Demandeur (RQSTR)

- Emetteur de cluster (CLUSDR)
- Récepteur de cluster (CLUSRCVR)

Un canal de transmission de messages est défini à l'aide d'un de ces types définis sur une extrémité et d'un type compatible à l'autre extrémité. Les combinaisons possibles sont les suivantes :

- Emetteur-récepteur
- Demandeur-serveur
- Demandeur-émetteur (rappel)
- Serveur-récepteur
- Emetteur de cluster-récepteur de cluster

Les instructions détaillées de création d'un canal émetteur-récepteur figurent dans [Définition des canaux](#). Pour consulter des exemples de paramètres requis pour la configuration des canaux émetteurs-récepteurs, consultez la rubrique [Exemples d'informations de configuration applicable à votre plateforme](#). Pour les paramètres requis pour la définition d'un canal de n'importe quel type, voir [DEFINE CHANNEL](#).

## Canaux émetteurs-récepteurs

Un émetteur sur un système démarre le canal de sorte qu'il puisse envoyer des messages à l'autre système. L'émetteur demande au récepteur, situé à l'autre extrémité du canal, de démarrer. L'émetteur envoie des messages de sa file d'attente de transmission au récepteur. Le récepteur place les messages dans la file d'attente de destination. [Figure 8](#), à la page 50 illustre cette situation.

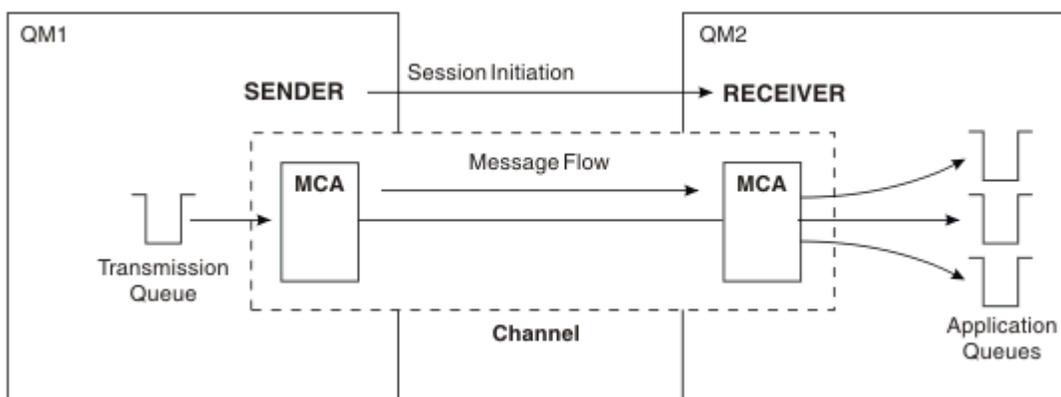


Figure 8. Un canal émetteur-récepteur

## Canaux demandeur-serveur

Un demandeur sur un système démarre le canal de sorte qu'il puisse recevoir des messages de l'autre système. Le demandeur demande au serveur, situé à l'autre extrémité du canal, de démarrer. Le serveur envoie des messages au demandeur à partir de la file d'attente de transmission définie dans sa définition de canal.

Un canal serveur peut également lancer la communication et envoyer des messages à un demandeur. Cela ne s'applique qu'aux serveurs *entièrement qualifiés*, à savoir les canaux serveur qui portent le nom de connexion du partenaire spécifié dans la définition de canal. Un serveur entièrement qualifié peut être démarré par un demandeur ou peut lancer une communication avec un demandeur.

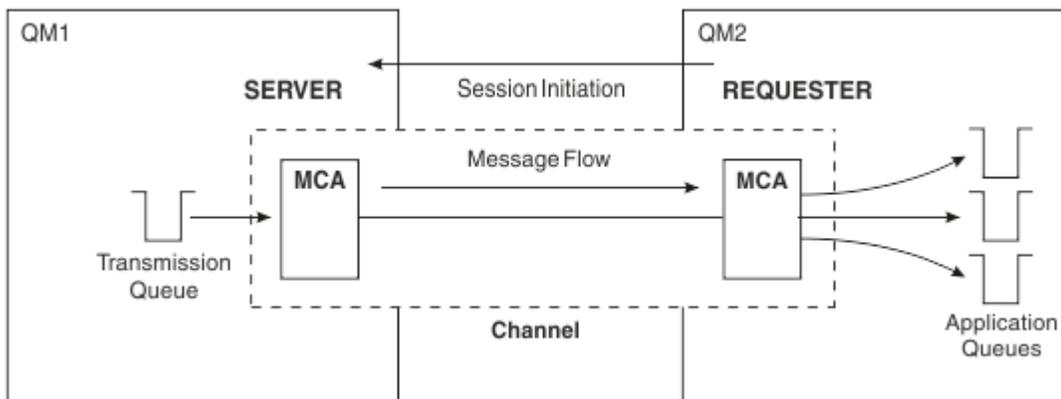


Figure 9. Un canal demandeur-serveur

### Canaux demandeur-émetteur

Le demandeur démarre le canal et l'émetteur met fin à l'appel. L'émetteur redémarre ensuite la communication en fonction des informations figurant dans sa définition de canal (connue sous le nom de *rappel*). Il envoie des messages de la file d'attente de transmission au demandeur.

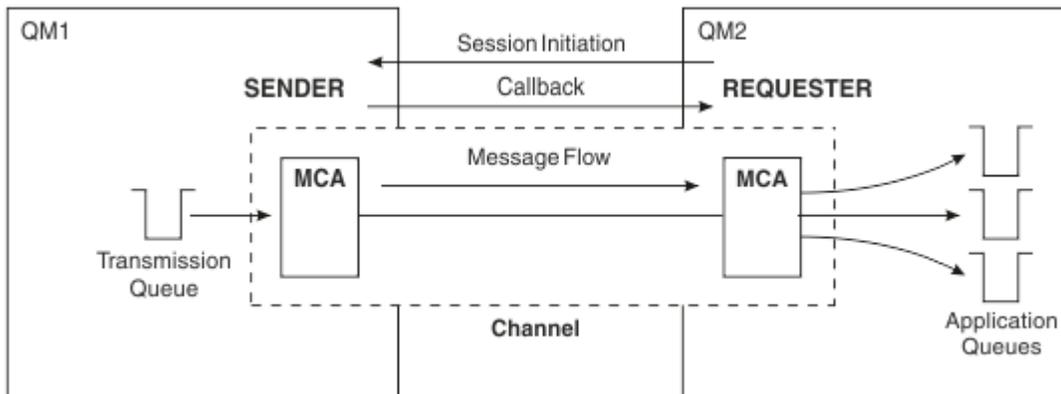


Figure 10. Un canal demandeur-émetteur

### Canaux serveur-récepteur

Ce type de canal est similaire au canal émetteur-récepteur mais s'applique uniquement aux serveurs *entièrement qualifiés*, à savoir les canaux serveur qui portent le nom de connexion du partenaire spécifié dans la définition de canal. Le démarrage du canal doit être déclenché à l'extrémité serveur de la liaison. L'illustration de ce processus est similaire à celle de la [Figure 8](#), à la page 50.

### Canaux émetteurs de cluster

Dans un cluster, chaque gestionnaire de files d'attente comporte un canal émetteur de cluster sur lequel il peut envoyer des informations de cluster à l'un des gestionnaires de files d'attente de référentiel complet. Les gestionnaires de files d'attente peuvent également envoyer des messages à d'autres gestionnaires de files d'attente sur des canaux émetteurs de cluster.

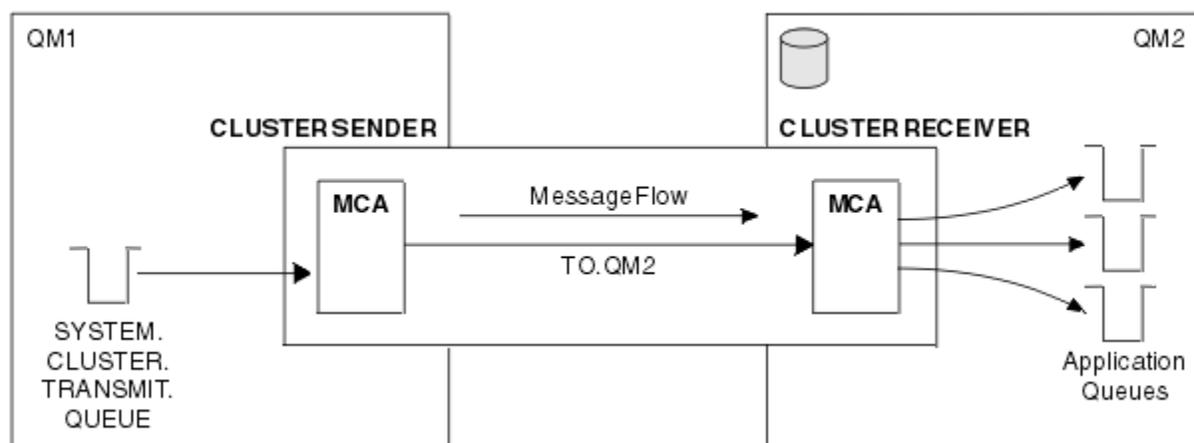


Figure 11. Un canal émetteur de cluster

## Canaux récepteurs de cluster

Dans un cluster, chaque gestionnaire de files d'attente comporte un canal récepteur de cluster sur lequel il peut recevoir des messages et des informations concernant le cluster. L'illustration de ce processus est similaire à celle de la [Figure 11](#), à la page 52.

## Files d'attente de rebut

La file d'attente de rebut (ou file d'attente des messages non livrés) est la file d'attente à laquelle des messages sont envoyés s'ils ne peuvent pas être acheminés vers la destination appropriée. Chaque gestionnaire de files d'attente a généralement une file d'attente de rebut.

Une *file d'attente de rebut* (DLQ), parfois appelée *file d'attente de messages non livrés*, est une file d'attente stockant des messages qui ne peuvent pas être livrés à leurs files d'attente de destination, par exemple parce que la file d'attente n'existe pas ou qu'elle est saturée. Les files d'attente de rebut sont également utilisées à l'extrémité d'émission d'un canal pour les erreurs de conversion de données. Chaque gestionnaire de files d'attente d'un réseau dispose généralement d'une file d'attente locale à utiliser comme file d'attente de rebut. Ainsi, les messages non distribuables à la destination appropriée pourront être stockés puis récupérés ultérieurement.

Les messages peuvent être placés dans la file d'attente de rebut par des gestionnaires de files d'attente, de 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. 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.

Il est conseillé de définir une file d'attente de rebut pour chaque gestionnaire de files d'attente. Si vous ne le faites pas et que l'agent MCA ne peut pas insérer un message, celui-ci reste dans la file d'attente de transmission et le canal s'arrête. De même, si des messages rapides non persistants (voir [Messages rapides non persistants](#)) ne peuvent pas être distribués et qu'aucune file d'attente de rebut n'existe sur le système cible, ces messages sont supprimés.

Cependant, les files d'attente de rebut peuvent avoir une incidence sur la séquence de distribution des messages et, par conséquent, vous pouvez choisir de ne pas les utiliser.

## Définition de files d'attente éloignées

Les définitions de file d'attente éloignée sont des définitions destinées aux files d'attente détenues par un autre gestionnaire de files d'attente.

Les applications peuvent extraire des messages uniquement des files d'attente locales, mais elles peuvent insérer des messages dans des files d'attente locales ou des files d'attente éloignées. Par conséquent, en sus d'une définition pour chacune de ses files d'attente locales, un gestionnaire de files d'attente peut comporter des *définitions de file d'attente éloignée*. L'avantage des définitions de file

d'attente éloignée est qu'elles permettent à une application d'insérer un message dans une file d'attente éloignée sans avoir à indiquer le nom de la file d'attente éloignée ou du gestionnaire de files d'attente éloignées ou le nom de la file d'attente de transmission. Grâce aux définitions de file d'attente éloignée, vous n'êtes pas tributaire des emplacements.

Les définitions de file d'attente éloignée peuvent être utilisées à d'autres fins, comme décrit ultérieurement.

## Mode d'accès au gestionnaire de files d'attente éloignées

Il se peut que vous n'ayez pas toujours un canal entre chaque gestionnaire de files d'attente source et cible. Il existe un certain nombre d'autres méthodes de liaison entre les deux, y compris le mode multi-tronçons, le partage de canaux, l'utilisation de canaux différents et la mise en cluster.

### Multi-tronçons

S'il n'y pas de liaison de communication directe entre le gestionnaire de files d'attente source et le gestionnaire de files d'attente cible, il est possible de traverser un ou plusieurs *gestionnaires de files d'attente intermédiaires* sur le chemin vers le gestionnaire de files d'attente cible. C'est ce qu'on appelle *multi-tronçons*.

Vous définissez des canaux entre tous les gestionnaires de files d'attente, ainsi que des files d'attente de transmission sur les gestionnaires de files d'attente intermédiaires. Cela est illustré dans la [Figure 12](#), à la page 53.

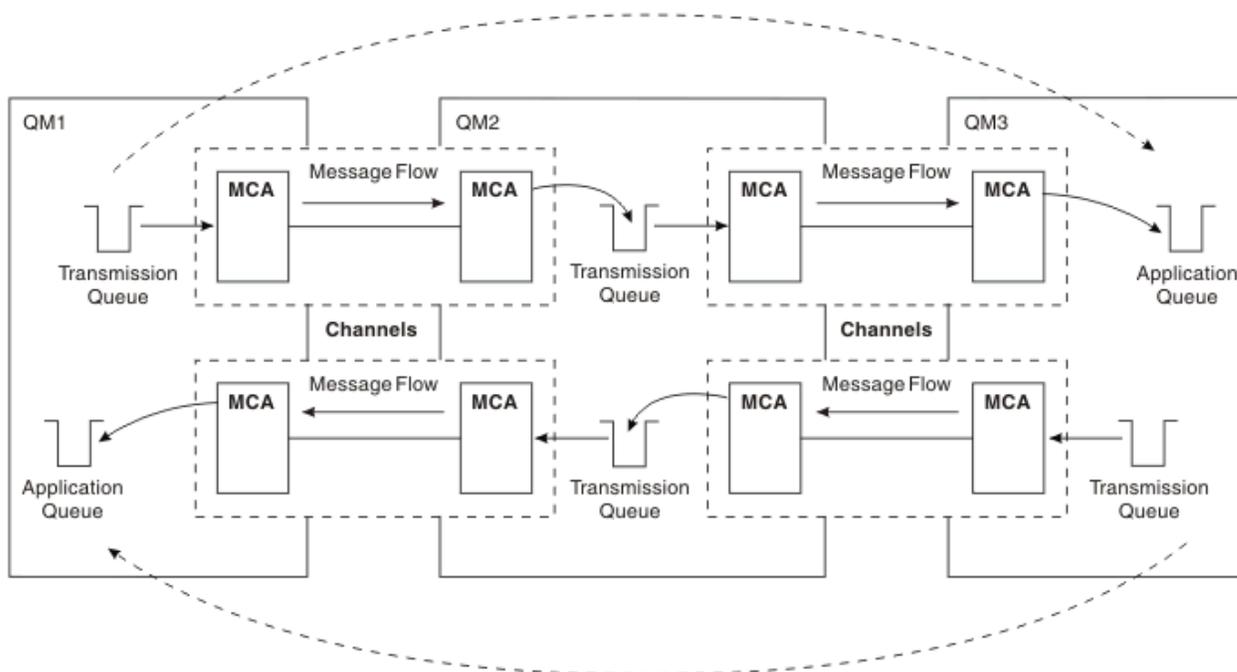


Figure 12. Traversée de gestionnaires de files d'attente intermédiaires

### Partage de canaux

En tant que concepteur d'application, vous avez le choix de forcer vos applications à spécifier le nom du gestionnaire de files d'attente éloignées ainsi que le nom de la file d'attente, ou de créer une *définition de file d'attente éloignée* pour chaque file d'attente éloignée. Cette définition contient le nom du gestionnaire de files d'attente éloignées, le nom de la file d'attente et le nom de la file d'attente de transmission.

Quelle que soit l'opération que vous choisissez, tous les messages provenant de toutes les applications adressant des files d'attente dans le même emplacement éloigné sont envoyés via la même file d'attente de transmission. Cela est illustré dans la [Figure 13](#), à la page 54.

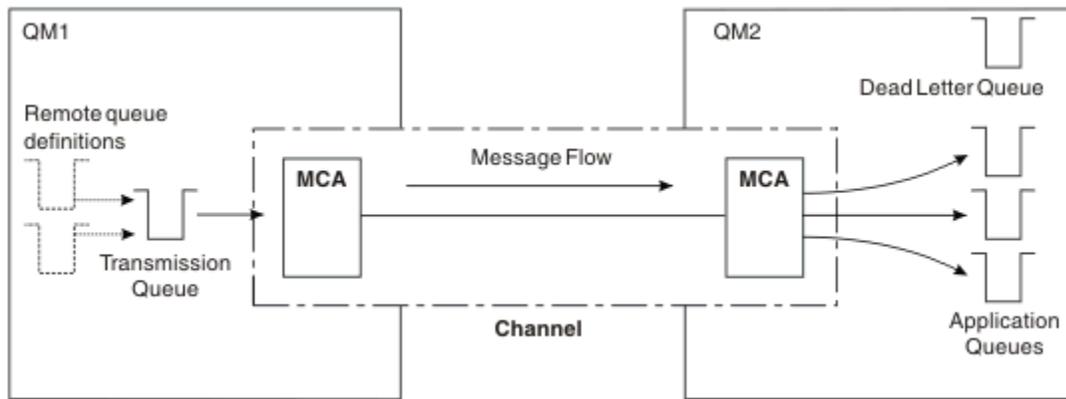


Figure 13. Partage d'une file d'attente de transmission

La Figure 13, à la page 54 indique que les messages envoyés de plusieurs applications à plusieurs files d'attente éloignées peuvent utiliser le même canal.

### Utilisation de canaux différents

Si vous avez des messages de types différents à envoyer entre deux gestionnaires de files d'attente, vous pouvez définir plusieurs canaux entre les deux. Vous aurez parfois besoin d'autres canaux, peut-être pour des raisons de sécurité ou pour sacrifier la vitesse de distribution au profit d'un trafic de messages extrêmement élevé.

Pour configurer un deuxième canal, vous devez définir un autre canal et une autre file d'attente de transmission, puis créer une définition de file d'attente éloignée indiquant l'emplacement et le nom de la file d'attente de transmission. Vos applications peuvent ensuite utiliser l'un ou l'autre des canaux mais les messages sont toujours distribués aux mêmes files d'attente cible. Cela est illustré dans la Figure 14, à la page 54.

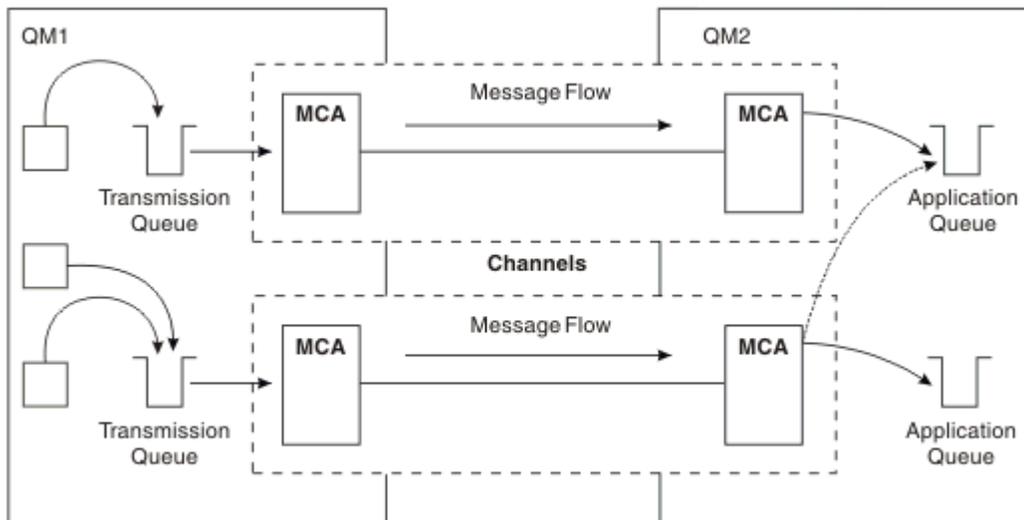


Figure 14. Utilisation de plusieurs canaux

Lorsque vous utilisez des définitions de file d'attente éloignée pour spécifier une file d'attente de transmission, vos applications ne doivent **pas** elles-mêmes spécifier l'emplacement (c-à-d, le gestionnaire de files d'attente de destination). Si elles le spécifient, le gestionnaire de files d'attente n'utilise pas les définitions de file d'attente éloignée. Grâce aux définitions de file d'attente éloignée, vous n'êtes pas tributaire des emplacements. Les applications peuvent insérer des messages dans une file d'attente *logique* sans connaître l'emplacement de la file d'attente et vous pouvez modifier la file d'attente *physique* sans avoir à modifier vos applications.

## Utilisation de la mise en cluster

Chaque gestionnaire de files d'attente dans un cluster définit un canal récepteur de cluster. Lorsqu'un autre gestionnaire de files d'attente souhaite envoyer un message à ce gestionnaire de files d'attente, il définit automatiquement le canal émetteur de cluster correspondant. Par exemple, si un cluster contient plusieurs instances d'une file d'attente, le canal émetteur de cluster peut être défini sur n'importe quel gestionnaire de files d'attente hébergeant la file d'attente. IBM MQ utilise un algorithme de gestion de charge de travail qui utilise une routine circulaire permettant de sélectionner un gestionnaire de files d'attente disponible de destination d'un message. Pour plus d'informations, voir [Clusters](#).

## Informations d'adressage

Lorsqu'une application insère des messages destinés à un gestionnaire de files d'attente éloignées, le gestionnaire de files d'attente local ajoute dans les messages un en-tête de transmission avant de les placer dans la file d'attente de transmission. Cet en-tête contient le nom de la file d'attente de destination et du gestionnaire de files d'attente, à savoir les *informations d'adressage*.

Dans un environnement comportant un gestionnaire de files d'attente unique, l'adresse d'une file d'attente de destination est établie lorsqu'une application ouvre une file d'attente dans laquelle insérer des messages. Dans la mesure où la file d'attente de destination se trouve sur le même gestionnaire de files d'attente, aucune information d'adressage n'est nécessaire.

Dans un environnement de mise en file d'attente répartie, le gestionnaire de files d'attente doit connaître non seulement le nom de la file d'attente de destination, mais également l'emplacement de cette file d'attente (c'est-à-dire, le nom du gestionnaire de files d'attente) et le chemin d'accès à cet emplacement éloigné (c'est-à-dire, la file d'attente de transmission). Ces informations d'adressage sont contenues dans l'en-tête de transmission. Le canal récepteur supprime l'en-tête de transmission et utilise les informations qu'il contient pour localiser la file d'attente de destination.

Vous pouvez éviter que vos applications aient à indiquer le nom du gestionnaire de files d'attente de destination si vous utilisez une définition de file d'attente éloignée. Cette définition indique le nom de la file d'attente éloignée, le nom du gestionnaire de files d'attente éloignées auquel des messages sont destinés, ainsi que le nom de la file d'attente de transmission servant à transférer les messages.

## Description des alias

Les alias servent à fournir une qualité de service pour les messages. L'alias de gestionnaire de files d'attente permet à un administrateur système de modifier le nom d'un gestionnaire de files d'attente cible sans que vous ayez à modifier vos applications. Il permet également à l'administrateur système de modifier le chemin d'accès à un gestionnaire de files d'attente de destination ou de configurer un chemin traversant un certain nombre d'autres gestionnaires de files d'attente (multi-tronçons). L'alias de file d'attente de réponses fournit une qualité de service pour les réponses.

Les alias de gestionnaire de files d'attente et les alias de file d'attente de réponses sont créés à l'aide d'une définition de file d'attente éloignée comportant une zone RNAME à blanc. Ces définitions ne définissent pas de files d'attente réelles ; elles sont utilisées par le gestionnaire de files d'attente pour résoudre des noms de file d'attente physique, des noms de gestionnaire de files d'attente et des files d'attente de transmission.

Les définitions d'alias se caractérisent par une zone RNAME à blanc.

## Résolution de nom de file d'attente

La résolution de nom de file d'attente a lieu sur tous les gestionnaires de files d'attente chaque fois qu'une file d'attente est ouverte. Elle a pour but d'identifier la file d'attente cible, le gestionnaire de files d'attente cible (qui peut être local) et le chemin d'accès à ce gestionnaire de files d'attente (qui peut être NULL). Le nom résolu comporte trois parties : le nom du gestionnaire de files d'attente, le nom de file d'attente et, si le gestionnaire de files d'attente est éloigné, la file d'attente de transmission.

Lorsqu'il existe une définition de file d'attente éloignée, aucune définition d'alias n'est référencée. Le nom de file d'attente fourni par l'application est résolu en nom de la file d'attente de destination,

du gestionnaire de files d'attente éloignées et de la file d'attente de transmission indiquée dans la définition de file d'attente éloignée. Pour plus d'informations sur la résolution de nom de file d'attente, voir [Résolution de nom de file d'attente](#).

S'il n'existe aucune définition de file d'attente éloignée et qu'un nom de gestionnaire de files d'attente est spécifié ou résolu par le service annuaire, le gestionnaire de files d'attente vérifie si une définition d'alias de gestionnaire de files d'attente correspond au nom de gestionnaire de files d'attente fourni. Si c'est le cas, les informations qu'elle contient sont utilisées pour résoudre le nom de gestionnaire de files d'attente en nom du gestionnaire de files d'attente de destination. La définition d'alias de gestionnaire de files d'attente peut également être utilisée pour déterminer la file d'attente de transmission sur le gestionnaire de files d'attente de destination.

Si le nom de file d'attente résolu ne correspond pas à une file d'attente locale, le nom de gestionnaire de files d'attente et le nom de file d'attente sont tous deux inclus dans l'en-tête de transmission de chaque message inséré par l'application dans la file d'attente de transmission.

La file d'attente de transmission généralement utilisée porte le même nom que le gestionnaire de files d'attente résolu, sauf si elle est modifiée par une définition de file d'attente éloignée ou une définition d'alias de gestionnaire de files d'attente. Si vous n'avez pas défini une file d'attente de transmission de ce type et que vous avez défini une file d'attente de transmission par défaut, cette dernière est alors utilisée.

 Les noms des gestionnaires de files d'attente exécutés sur z/OS sont limités à quatre caractères.

## Définition des alias de gestionnaire de files d'attente

Les définitions d'alias de gestionnaire de files d'attente s'appliquent lorsqu'une application ouvrant une file d'attente pour insérer un message indique le nom de file d'attente **et** le nom de gestionnaire de files d'attente.

Les définitions d'alias de gestionnaire de files d'attente sont utilisées pour les trois opérations suivantes :

- Lors de l'envoi de messages, pour remapper le nom de gestionnaire de files d'attente
- Lors de l'envoi de messages, pour modifier ou spécifier la file d'attente de transmission
- Lors de la réception de messages, pour déterminer si le gestionnaire de files d'attente local est la destination prévue de ces messages

## Messages sortants - remappage du nom de gestionnaire de files d'attente

Les définitions d'alias de gestionnaire de files d'attente permettent de remapper le nom de gestionnaire de files d'attente indiqué dans un appel MQOPEN. Par exemple, un appel MQOPEN spécifie le nom de file d'attente THISQ et le nom de gestionnaire de files d'attente YOURQM. Sur le gestionnaire de files d'attente local, il existe une définition d'alias de gestionnaire de files d'attente similaire à l'exemple suivant :

```
DEFINE QREMOTE (YOURQM) RQMNAME(REALQM)
```

Cet exemple montre que le gestionnaire de files d'attente réel à utiliser, lorsqu'une application insère des messages dans le gestionnaire de files d'attente YOURQM, est REALQM. Si le gestionnaire de files d'attente local est REALQM, il insère les messages dans la file d'attente THISQ, qui est une file d'attente locale. Si le gestionnaire de files d'attente local n'est pas intitulé REALQM, il achemine le message vers une file d'attente de transmission intitulée REALQM. Le gestionnaire de files d'attente modifie l'en-tête de transmission de sorte qu'il indique REALQM à la place de YOURQM.

## Messages sortants - modification ou spécification de la file d'attente de transmission

La [Figure 15](#), à la [page 57](#) présente un scénario où les messages arrivent sur le gestionnaire de files d'attente QM1 à l'aide d'en-têtes de transmission affichant des noms de file d'attente sur le gestionnaire de files d'attente QM3. Dans ce scénario, QM3 est accessible à l'aide de plusieurs tronçons via QM2.

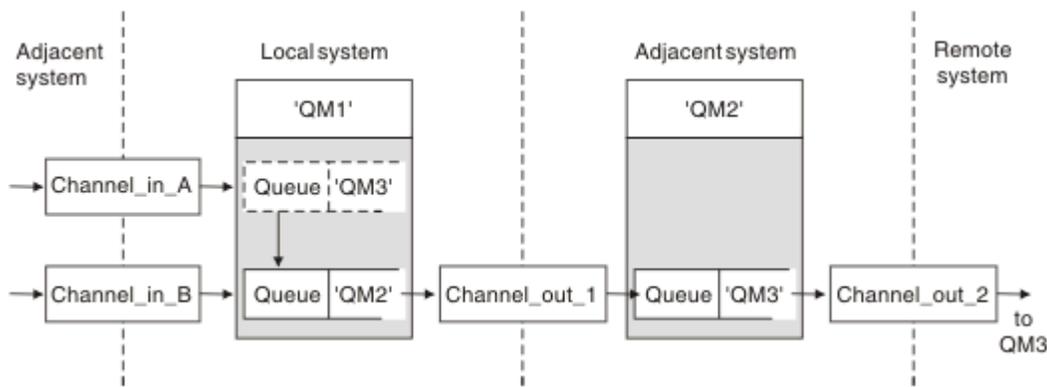


Figure 15. Alias de gestionnaire de files d'attente

Tous les messages de QM3 sont capturés sur QM1 à l'aide d'un alias de gestionnaire de files d'attente. Ce dernier est intitulé QM3 et contient la définition QM3 via la file d'attente de transmission QM2. La définition est similaire à l'exemple suivant :

```
DEFINE QREMOTE (QM3) RNAME(' ') RQMNAME(QM3) XMITQ(QM2)
```

Le gestionnaire de files d'attente insère les messages dans la file d'attente de transmission QM2 mais ne modifie pas l'en-tête de la file d'attente de transmission car le nom du gestionnaire de files d'attente de destination, QM3, ne change pas.

Tous les messages arrivant sur QM1 et présentant un en-tête de transmission contenant un nom de file d'attente sur QM2 sont également insérés dans la file d'attente de transmission QM2. Les messages comportant des destinations différentes sont ainsi collectés sur une file d'attente de transmission commune d'un système adjacent approprié pour être retransmis ensuite à leurs destinations.

## Messages entrants - détermination de la destination

Un agent MCA récepteur ouvre la file d'attente référencée dans l'en-tête de transmission. S'il existe une définition d'alias de gestionnaire de files d'attente du même nom que le gestionnaire de files d'attente référencé, le nom de gestionnaire de files d'attente reçu dans l'en-tête de transmission est remplacé par le nom RQMNAME issu de cette définition.

Ce processus est utilisé pour les deux opérations suivantes :

- Acheminement des messages vers un autre gestionnaire de files d'attente
- Modification du nom de gestionnaire de files d'attente pour qu'il soit identique à celui du gestionnaire de files d'attente local

## Définition des alias de files d'attente de réponse

Une définition d'alias de file d'attente de réponses indique d'autres noms pour les informations de réponse dans le descripteur de message. Cela a pour avantage que vous pouvez modifier le nom d'une file d'attente ou d'un gestionnaire de files d'attente sans avoir à modifier vos applications.

## Résolution de nom de file d'attente

Lorsqu'une application répond à un message, elle utilise les données du *descripteur* du message qu'elle a reçu pour trouver le nom de la file d'attente à laquelle répondre. L'application émettrice indique l'emplacement dans lequel les réponses sont envoyées et joint ces informations à ses messages. Ce concept doit être coordonné dans le cadre de votre conception d'application.

La résolution de nom de file d'attente a lieu à l'extrémité émettrice de votre application, avant que le message soit inséré dans une file d'attente. La résolution de nom de file d'attente se produit par

conséquent avant une interaction avec l'application distante vers laquelle le message est envoyé. C'est la seule situation où la résolution de nom a lieu lorsqu'une file d'attente n'est pas ouverte.

## Résolution de nom de file d'attente à l'aide d'un alias de gestionnaire de files d'attente

En général, une application spécifie une file d'attente de réponses et laisse le nom du gestionnaire de files d'attente de réponses à blanc. Le gestionnaire de files d'attente complète son propre nom au moment de l'insertion. Cette méthode fonctionne correctement sauf lorsque vous souhaitez qu'un autre canal soit utilisé pour les réponses, par exemple un canal utilisant la file d'attente de transmission QM1\_relief à la place du canal de retour par défaut qui utilise la file d'attente de transmission QM1. Dans cette situation, les noms de gestionnaire de files d'attente indiqués dans les en-têtes de file d'attente de transmission ne correspondent pas à des noms de gestionnaire de files d'attente "réels", mais sont spécifiés à l'aide des définitions d'alias de gestionnaire de files d'attente. Pour renvoyer des réponses via des chemins secondaires, il est également nécessaire de mapper les données de file d'attente de réponses à l'aide des définitions d'alias de file d'attente de réponses.

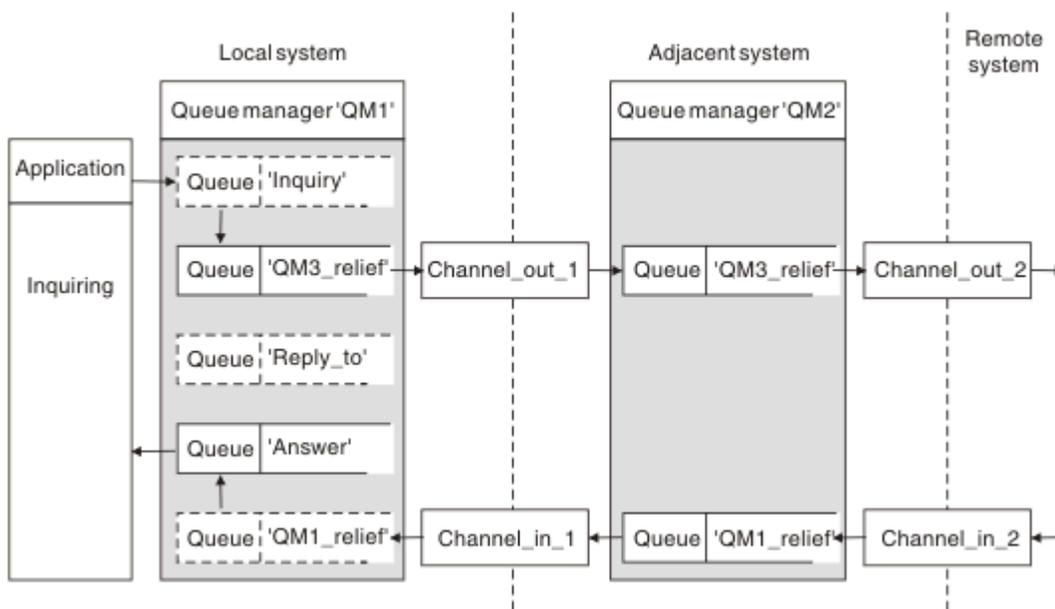


Figure 16. Alias de file d'attente de réponses utilisé pour la modification de l'emplacement des réponses

Dans l'exemple de la figure [Figure 16](#), à la page 58 :

1. L'application insère un message à l'aide de l'appel MQPUT et en spécifiant les informations suivantes dans le descripteur de message :

```
ReplyToQ='Reply_to'
ReplyToQMgr=''
```

ReplyToQMgr doit être à blanc pour que l'alias de file d'attente de réponses soit utilisé.

2. Vous créez une définition d'alias de file d'attente de réponses intitulée Reply\_to, qui contient le nom Answer, ainsi que le nom de gestionnaire de files d'attente QM1\_relief.

```
DEFINE QREMOTE ('Reply_to') RNAME ('Answer')
RQMNAME ('QM1_relief')
```

3. Les messages sont envoyés à l'aide d'un descripteur de message affichant ReplyToQ='Answer' et ReplyToQMgr='QM1\_relief'.

4. La spécification d'application doit inclure l'information selon laquelle les réponses doivent être trouvées dans la file d'attente Answer et non dans Reply\_to.

Pour préparer les réponses, vous devez créer le canal de retour parallèle, en définissant :

- Sur QM2, la file d'attente de transmission intitulée QM1\_relief

```
DEFINE QLOCAL ('QM1_relief') USAGE(XMITQ)
```

- Sur QM1, l'alias de gestionnaire de files d'attente QM1\_relief

```
DEFINE QREMOTE ('QM1_relief') RNAME() RQMNAME(QM1)
```

Cet alias de gestionnaire de files d'attente met fin à la chaîne des canaux de retour parallèles et capture les messages pour QM1.

Si vous pensez que vous serez amené à effectuer cette opération ultérieurement, assurez-vous que les applications utilisent le nom d'alias depuis le début. Pour l'instant, il s'agit d'un alias de file d'attente normal pour la file d'attente de réponses mais, par la suite, il peut être remplacé par un alias de gestionnaire de files d'attente.

## Nom de file d'attente de réponses

Des précautions particulières sont nécessaires lors de l'attribution de nom aux files d'attente de réponses. La raison pour laquelle une application insère un nom de file d'attente de réponses dans le message est qu'elle peut spécifier la file d'attente à laquelle ses réponses sont envoyées. Lors de la création d'une définition d'alias de file d'attente de réponses à l'aide de ce nom, la file d'attente de réponses réelle (c-à-d, une définition de file d'attente locale) ne peut pas porter le même nom. Par conséquent, la définition d'alias de file d'attente de réponses doit contenir un nouveau nom de file d'attente ainsi que le nom de gestionnaire de files d'attente, et la spécification d'application doit inclure l'information selon laquelle ses réponses se trouvent dans cette autre file d'attente.

Les applications doivent désormais extraire les messages d'une file d'attente différente de celle qu'elles ont désignée comme file d'attente de réponses, lorsqu'elles insèrent le message d'origine.

## Composants de cluster

Les clusters sont composés de gestionnaires de files d'attente, de référentiels de cluster, de canaux de cluster et de files d'attente de cluster.

Voir les sous-rubriques suivantes pour plus d'informations sur chaque composant du cluster :

### Concepts associés

[Comparaison de la mise en cluster et de la mise en file d'attente répartie](#)

### Tâches associées

[Configuration d'un cluster de gestionnaires de files d'attente](#)

[Configuration d'un nouveau cluster](#)

## Référentiel de cluster

Un référentiel est un ensemble d'informations concernant les gestionnaires de files d'attente membres d'un cluster.

Les informations du référentiel comprennent les noms des gestionnaires de files d'attente, leurs emplacements, leurs canaux, les files d'attente qu'ils hébergent et d'autres détails. Les informations sont stockées sous forme de messages dans une file d'attente appelée

SYSTEM.CLUSTER.REPOSITORY.QUEUE. Cette file d'attente est l'un des objets par défaut.  Sous [Multiplateformes](#), il est défini lors de la création d'un gestionnaire de files d'attente IBM

MQ.  Dans IBM MQ for z/OS, elle est définie dans le cadre de la personnalisation du gestionnaire de files d'attente.

## Référentiel complet et référentiel partiel

En règle générale, deux gestionnaires de files d'attente dans un cluster contiennent un référentiel complet. Les autres gestionnaires de files d'attente contiennent tous un référentiel partiel.

Un gestionnaire de files d'attente qui héberge un ensemble complet d'informations concernant tous les gestionnaires de files d'attente d'un cluster dispose d'un référentiel complet. Les autres gestionnaires de files d'attente du cluster comportent des référentiels partiels contenant un sous-ensemble des informations des référentiels complets.

Un référentiel partiel contient uniquement des informations relatives aux gestionnaires avec lesquels le gestionnaire de files d'attente a besoin d'échanger des messages. Les gestionnaires de files d'attente demandent les mises à jour apportées aux informations dont ils ont besoin pour que, si ces informations sont modifiées, le gestionnaire de files d'attente de référentiel complet leur envoie les nouvelles informations. La plupart du temps, le référentiel partiel contient toutes les informations nécessaires pour qu'un gestionnaire de files d'attente soit opérationnel dans le cluster. Lorsqu'un gestionnaire de files d'attente a besoin d'informations complémentaires, il interroge le référentiel complet et met à jour son référentiel partiel. Les gestionnaires de files d'attente utilisent la file d'attente `SYSTEM.CLUSTER.COMMAND.QUEUE` pour demander des mises à jour aux référentiels et les recevoir.

Lorsque vous migrez des gestionnaires de files d'attente qui sont membres d'un cluster, migrez les référentiels complets avant les référentiels partiels. En effet, un référentiel plus ancien ne peut pas stocker des attributs récents introduits dans une nouvelle édition. Il les tolère mais ne les stocke pas.

## Gestionnaire de files d'attente de cluster

Un gestionnaire de files d'attente de cluster est un gestionnaire de files d'attente membre d'un cluster.

Un gestionnaire de files d'attente peut être membre de plusieurs clusters. Chaque gestionnaire de files d'attente de cluster doit porter un nom unique dans tous les clusters dont il est membre.

Un gestionnaire de files d'attente de cluster peut héberger des files d'attente qu'il annonce aux autres gestionnaires de files d'attente du cluster. Ce n'est cependant pas obligatoire. Il peut plutôt envoyer des messages dans les files d'attente hébergées ailleurs dans le cluster et recevoir uniquement les réponses qui sont explicitement dirigées vers ce gestionnaire.

 Dans IBM MQ for z/OS, un gestionnaire de files d'attente de cluster peut être membre d'un groupe de partage de files d'attente. Dans ce cas, il partage ses définitions de file d'attente avec les autres gestionnaires de files d'attente du même groupe de partage.

Les gestionnaires de files d'attente de cluster sont autonomes. Ils disposent d'un contrôle complet sur les files d'attente et les canaux qu'ils définissent. Leurs définitions ne peuvent pas être modifiées par d'autres gestionnaires de files d'attente (autres que les gestionnaires du même groupe de partage de files d'attente). Les gestionnaires de files d'attente de référentiel ne contrôlent pas les définitions des autres gestionnaires de files d'attente du cluster. Ils contiennent un ensemble complet de toutes les définitions pour les utiliser si nécessaire. Un cluster est une fédération de gestionnaires de files d'attente.

Après vous avez créé ou modifié une définition sur un gestionnaire de files d'attente de cluster, les informations sont envoyées au gestionnaire de files d'attente de référentiel complet. Les autres référentiels du cluster sont mis à jour ultérieurement.

## Gestionnaire de files d'attente de référentiel complet

Un gestionnaire de files d'attente de référentiel complet est un gestionnaire de files d'attente de cluster qui contient une représentation complète des ressources du cluster. Pour garantir la disponibilité, configurez plusieurs gestionnaires de files d'attente de référentiel complet dans chaque cluster. Les gestionnaires de files d'attente de référentiel complet reçoivent des informations envoyées par les autres

gestionnaires de files d'attente du cluster et mettent à jour leurs référentiels. Ils s'envoient entre eux des messages pour s'assurer de disposer des nouvelles informations à jour concernant le cluster.

## Gestionnaires de files d'attente et référentiels

Chaque cluster comprend au moins un (de préférence, deux) gestionnaire de files d'attente contenant des référentiels complet des informations sur les gestionnaires de files d'attente, les files d'attente et les canaux d'un cluster. Ces référentiels contiennent également les demandes mises à jour des informations émanant des autres gestionnaires de files d'attente du cluster.

Les autres gestionnaires de files d'attente comportent un référentiel partiel contenant des informations sur le sous-ensemble de files d'attente et de gestionnaire de files d'attente avec lesquels ils doivent communiquer. Les gestionnaires de files d'attente génèrent leurs référentiels partiels en effectuant des interrogations lorsqu'ils ont besoin pour la première fois d'accéder à une autre file d'attente ou un autre gestionnaire de files d'attente. Ils demandent à être avertis des nouvelles informations concernant cette file d'attente ou ce gestionnaire de files d'attente.

Chaque gestionnaire de files d'attente stocke ses informations de référentiel dans des messages dans une file d'attente appelée `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. Les gestionnaires de files d'attente échangent des informations de référentiel dans des messages dans une file d'attente appelée `SYSTEM.CLUSTER.COMMAND.QUEUE`.

Chaque gestionnaire de files d'attente qui rejoint un cluster définit un canal émetteur de cluster, `CLUSSDR` vers l'un des référentiels. Il prend immédiatement connaissance des autres gestionnaires de files d'attente du cluster qui contiennent des référentiels complets. Il peut alors demander des informations à n'importe lequel des référentiels. Lorsque le gestionnaire de files d'attente envoie des informations au référentiel choisi, il envoie également les informations à une autre référentiel (s'il en existe un).

Un référentiel complet est mis à jour lorsque le gestionnaire de files d'attente qui l'héberge reçoit de nouvelles informations d'un des gestionnaires de files d'attente qui lui sont associés. Les nouvelles informations sont également envoyées à un autre référentiel pour réduire le risque de délai si un gestionnaire de files d'attente de référentiel est hors service. Comme toutes les informations sont envoyées deux fois, les référentiels doivent supprimer les doublons. Chaque élément d'information est associé à un numéro de séquence que les référentiels utilisent pour identifier les doublons. Tous les référentiels se synchronisent entre eux en échangeant des messages.

## Files d'attente de cluster

Une file d'attente de cluster est une file d'attente hébergée par un gestionnaire de files d'attente de cluster et accessible aux autres gestionnaires de files d'attente dans le cluster.

Une file d'attente de cluster est annoncée aux autres gestionnaires de files d'attente de cluster. Ces autres gestionnaires de files d'attente de cluster peuvent insérer des messages dans une file d'attente de cluster sans qu'une définition de file d'attente éloignée correspondante soit nécessaire. Une file d'attente de cluster peut être annoncée dans plusieurs clusters à l'aide d'une liste de noms de cluster.

Lorsqu'une file d'attente est annoncée, les gestionnaires de files d'attente de cluster peuvent y insérer des messages. Pour insérer un message, le gestionnaire de files d'attente doit déterminer, à partir des référentiels complets, l'emplacement où la file d'attente est hébergée. Il ajoute alors des informations de routage au message et insère ce dernier dans une file d'attente de transmission du cluster.

 Une file d'attente de cluster peut être une file d'attente partagée par les membres d'un groupe de partage de files d'attente dans IBM MQ for z/OS.

### Tâches associées

[Définition de files d'attente de cluster](#)

## Comparaison entre les files d'attente partagées et les files d'attente de cluster

Ces informations vous permettent de comparer les files d'attente partagées et les files d'attente de cluster, et de sélectionner le type de file d'attente le plus approprié pour votre système.

### Coûts de l'initiateur de canal

Etant donné que les messages sont envoyés par les canaux, vous devez prévoir les coûts de l'initiateur de canal en plus des coûts de l'application. Le réseau comporte des coûts car les canaux reçoivent et placent des messages. Ces coûts ne sont pas inclus dans les files d'attente partagées, qui utilisent par conséquent une puissance de traitement inférieure à celle des files d'attente de cluster lors du transfert des messages entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente.

### Disponibilité des messages

Lors de l'insertion dans une file d'attente, les files d'attente de cluster envoient le message à l'un des gestionnaires de files d'attente disposant de canaux actifs qui sont connectés à votre gestionnaire de files d'attente. Dans le gestionnaire de files d'attente éloignées, lorsque les applications utilisées pour le traitement des messages ne fonctionnent pas, les messages ne sont pas traités et attendent le démarrage des applications. De même, si un gestionnaire de files d'attente est arrêté, tous les messages dans le gestionnaire de files d'attente sont indisponibles jusqu'au redémarrage du gestionnaire de files d'attente. La disponibilité des messages pour ces instances est inférieure à celle lors de l'utilisation des files d'attente partagées.

Lorsque vous utilisez des files d'attente partagées, les applications dans le groupe de partage de files d'attente peuvent recevoir les messages qui sont envoyés. Si vous arrêtez un des gestionnaires de files d'attente du groupe de partage de files d'attente, les messages sont disponibles pour les autres gestionnaires de files d'attente, fournissant ainsi une disponibilité de message plus élevée que lors de l'utilisation des files d'attente de cluster.

### Capacité

Le coût d'une unité de couplage est plus élevé que celui d'un disque ; par conséquent, le coût de stockage de 1 000 000 messages dans une file d'attente locale est inférieur à celui d'une unité de couplage ayant une capacité suffisante pour le stockage du même nombre de messages.

### Envoi aux autres gestionnaires de files d'attente

Les messages de file d'attente partagée sont uniquement disponibles dans un groupe de partage de files d'attente. Si vous voulez utiliser un gestionnaire de files d'attente hors du groupe de partage de files d'attente, vous devez le faire à l'aide des canaux. Vous pouvez utiliser la mise en cluster pour équilibrer la charge de travail entre plusieurs gestionnaires de files d'attente réparties éloignées.

### Equilibrage de la charge

La mise en cluster peut être utilisée pour identifier les canaux et gestionnaires de files d'attente qui vont recevoir une partie des messages envoyés. Par exemple, vous pouvez envoyer 60 % des messages à un gestionnaire de files d'attente et 40 % à un autre gestionnaire de files d'attente. Cette instance ne dépend pas de la capacité d'exécution des travaux du gestionnaire de files d'attente éloignées. Le système lié au premier gestionnaire de files d'attente peut être surchargé et le système lié au second gestionnaire peut être en mode veille, mais la plupart des messages seront dirigés vers le premier gestionnaire de files d'attente.

Grâce aux files d'attente partagées, deux systèmes CICS peuvent recevoir des messages. Si un système est surchargé, l'autre système prend en charge presque toute la charge de travail.

## Canaux de cluster

Dans chaque référentiel complet, vous définissez manuellement un canal récepteur de cluster et un ensemble de canaux émetteurs de cluster pour vous connecter à tout autre référentiel complet dans le cluster. Lorsque vous ajoutez un référentiel partiel, vous définissez manuellement un canal récepteur de cluster et un canal émetteur de cluster unique qui se connecte à l'un des référentiels complets. D'autres canaux émetteurs de cluster sont définis automatiquement par le cluster, le cas échéant. Les canaux émetteurs de cluster définis automatiquement tirent leurs attributs de la définition de canal récepteur de cluster correspondante dans le gestionnaire de files d'attente de réception.

### Canal récepteur de cluster : CLUSRCVR

Une définition de canal CLUSRCVR définit l'extrémité d'un canal, sur laquelle un gestionnaire de files d'attente de cluster peut recevoir des messages d'autres gestionnaires de files d'attente du cluster.

Vous devez définir au moins un canal CLUSRCVR pour chaque gestionnaire de files d'attente de cluster. En définissant le canal CLUSRCVR, le gestionnaire de files d'attente indique aux autres gestionnaires qu'il est disponible pour recevoir des messages.

Une définition de canal CLUSRCVR permet également aux autres gestionnaires de files d'attente de définir automatiquement les définitions de canal émetteur de cluster correspondantes. Voir la section [«Canaux émetteurs de cluster définis automatiquement»](#), à la page 63 de cet article.

### Canal émetteur de cluster : CLUSSDR

Vous définissez manuellement un canal CLUSSDR entre chaque gestionnaire de files d'attente de référentiel complet et tout autre gestionnaire de files d'attente de référentiel complet dans le cluster. Toutes les mises à jour échangées par les référentiels complets sont transmises exclusivement sur ces canaux. En définissant manuellement ces canaux, vous contrôlez explicitement le réseau de référentiels complets.

Lorsque vous ajoutez un gestionnaire de files d'attente de référentiel partiel à un cluster, vous définissez manuellement un canal CLUSSDR unique qui se connecte à l'un des référentiels complets. Peu importe le référentiel complet choisi car une fois le premier contact établi, d'autres objets de gestionnaire de files d'attente de cluster, canaux CLUSSDR compris, sont définis automatiquement le cas échéant. Cela permet au gestionnaire de files d'attente d'envoyer des informations de cluster à n'importe quel référentiel complet et d'envoyer des messages à n'importe quel gestionnaire de files d'attente dans le cluster.

Comme indiqué dans la section de cet article, les canaux émetteurs définis automatiquement sont basés sur la configuration du canal récepteur de cluster. Ainsi, les propriétés de canal que vous définissez sur les canaux de cluster doivent être identiques sur les canaux CLUSSDR et les canaux récepteurs de cluster correspondants, ou définissez uniquement des canaux récepteurs de cluster.

Les canaux CLUSSDR doivent être définis manuellement uniquement pour les raisons indiquées précédemment. C'est-à-dire dans le but de connecter initialement un référentiel partiel à un référentiel complet ou de connecter entre eux deux référentiels complets. La configuration manuelle d'un canal CLUSSDR qui se connecte à un référentiel partiel ou à un gestionnaire de files d'attente qui n'est pas dans le cluster génère des messages d'erreur comme AMQ9427 et AMQ9428. Même s'il peut s'agir parfois d'une situation temporaire inévitable, par exemple, lors de la modification de l'emplacement d'un référentiel complet, la définition manuelle doit être supprimée dès que possible.

### Canaux émetteurs de cluster définis automatiquement

Généralement, lorsque vous ajoutez un gestionnaire de files d'attente de référentiel complet à un cluster, vous définissez manuellement deux canaux de cluster seulement sur le gestionnaire de files d'attente :

- Un canal émetteur de cluster (CLUSSDR) à un gestionnaire de files d'attente de référentiel complet pour le cluster.
- Un canal récepteur de cluster (CLUSRCVR).

Le canal CLUSSDR que vous définissez permet au gestionnaire de files d'attente d'établir le premier contact avec le cluster. Une fois le premier contact établi, d'autres canaux CLUSSDR sont définis automatiquement par le cluster, le cas échéant.

Un canal CLUSSDR défini automatiquement tire ses attributs de la définition de canal CLUSRCVR correspondante sur le gestionnaire de files d'attente de réception. Même s'il existe un canal CLUSSDR défini manuellement, les attributs du canal CLUSSDR défini automatiquement sont utilisés. Supposons par exemple que vous définissez un canal CLUSRCVR sans indiquer de numéro de port dans le paramètre **CONNNAME** et que vous définissez manuellement un canal CLUSSDR qui n'indique pas de numéro de port. Lorsque le canal CLUSSDR défini automatiquement remplace celui qui est défini manuellement, le numéro de port (tiré du canal CLUSRCVR) reste à blanc. Le numéro de port par défaut est utilisé et le canal échoue.

Lorsqu'il existe des différences de configuration entre un canal CLUSSDR défini manuellement et une définition de canal CLUSRCVR correspondante, certaines prennent effet immédiatement (par exemple les paramètres d'équilibrage de charge) et d'autres prennent effet uniquement lors du redémarrage du canal (par exemple, la configuration TLS).

Pour éviter toute confusion, respectez si possible les consignes suivantes :

- Définissez manuellement des canaux CLUSSDR uniquement dans le but de pointer vers des référentiels complets.
- Après avoir défini manuellement des canaux CLUSSDR, configurez-les de sorte qu'ils soient identiques à la définition de canal CLUSRCVR correspondante sur le gestionnaire de files d'attente de réception.

Voir aussi [Utilisation des canaux définis automatiquement](#).

### **Concepts associés**

[Utilisation de canaux définis automatiquement](#)

[Utilisation de files d'attente de transmission de cluster et de canaux émetteurs de cluster](#)

### **Tâches associées**

[Configuration d'un nouveau cluster](#)

[Ajout d'un gestionnaire de files d'attente à un cluster](#)

## **Rubriques de cluster**

Les rubriques de cluster sont des rubriques d'administration avec l'attribut **cluster** défini. Des informations les concernant sont transmises à tous les membres d'un cluster et associées aux rubriques locales pour créer des portions d'un espace de sujet qui s'étend sur plusieurs gestionnaires de files d'attente. Cela permet aux messages publiés dans une rubrique sur un gestionnaire de files d'attente d'être distribués aux abonnements d'autres gestionnaires de files d'attente du cluster.

Lorsque vous définissez une rubrique de cluster dans un gestionnaire de files d'attente, la définition de la rubrique de cluster est envoyée aux gestionnaires de files d'attente des référentiels complets. Ces derniers propagent ensuite la définition de rubrique en cluster à tous les gestionnaires de files d'attente du cluster, ce qui rend la même rubrique de cluster disponible pour les diffuseurs de publications et les abonnés dans un gestionnaire de files d'attente du cluster. Le gestionnaire de files d'attente sur lequel vous créez une rubrique de cluster est appelé hôte de rubrique de cluster. La rubrique de cluster peut être utilisée par n'importe quel gestionnaire de files d'attente du cluster, mais les modifications à lui apporter doivent être effectuées sur celui où elle est définie (l'hôte). Ces modifications sont ensuite propagées à tous les membres du cluster par le biais des référentiels complets.

Pour toute information sur la configuration des rubriques de cluster de sorte qu'elles utilisent le *routing direct* ou le *routing via un hôte de rubrique* et sur l'héritage des rubriques de cluster et les abonnements génériques, voir [Définition de rubriques de cluster](#).

Pour des informations sur les commandes à utiliser pour afficher les rubriques de cluster, voir les informations connexes.

### **Concepts associés**

[Utilisation des rubriques d'administration](#)

## Référence associée

[DISPLAY TOPIC](#)

[DISPLAY TPSTATUS](#)

[DISPLAY SUB](#)

## Objets de cluster par défaut

 Sur Multiplatforms, les objets de cluster par défaut sont inclus dans l'ensemble d'objets par défaut créé automatiquement lorsque vous définissez un gestionnaire de files d'attente.  Sous z/OS, vous trouverez les définitions d'objet de cluster par défaut dans les exemples de personnalisation.

**Remarque :** Vous pouvez modifier les définitions de canal par défaut de la même manière que toute autre définition de canal, en exécutant des commandes MQSC ou PCF. Ne modifiez pas les définitions de files d'attente par défaut, sauf `SYSTEM.CLUSTER.HISTORY.QUEUE`.

### **SYSTEM.CLUSTER.COMMAND.QUEUE**

Chaque gestionnaire de files d'attente d'un cluster dispose d'une file d'attente locale appelée `SYSTEM.CLUSTER.COMMAND.QUEUE` qui est utilisée pour transférer des messages au référentiel complet. Les messages contiennent les informations nouvelles ou modifiées concernant le gestionnaire de files d'attente, ou les demandes d'informations à propos d'autres gestionnaires de files d'attente. `SYSTEM.CLUSTER.COMMAND.QUEUE` est vide normalement.

### **SYSTEM.CLUSTER.HISTORY.QUEUE**

Chaque gestionnaire de files d'attente d'un cluster dispose d'une file d'attente locale appelée `SYSTEM.CLUSTER.HISTORY.QUEUE`. `SYSTEM.CLUSTER.HISTORY.QUEUE` est utilisée pour le stockage de l'historique des informations d'état de cluster pour les opérations de maintenance.

Dans les paramètres d'objet par défaut, `SYSTEM.CLUSTER.HISTORY.QUEUE` est défini sur `PUT (ENABLED)`. Pour supprimer la collecte d'historique, remplacez le paramètre par `PUT (DISABLED)`.

### **SYSTEM.CLUSTER.REPOSITORY.QUEUE**

Chaque gestionnaire de files d'attente d'un cluster dispose d'une file d'attente locale appelée `SYSTEM.CLUSTER.REPOSITORY.QUEUE`. Cette file d'attente est utilisée pour stocker toutes les informations de référentiel complet. Cette file d'attente n'est pas vide normalement.

### **SYSTEM.CLUSTER.TRANSMIT.QUEUE**

Chaque gestionnaire de files d'attente dispose d'une file d'attente locale appelée `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. `SYSTEM.CLUSTER.TRANSMIT.QUEUE` est la file d'attente de transmission par défaut pour tous les messages vers toutes les files d'attente et tous les gestionnaires de files d'attente se trouvant dans des clusters. Vous pouvez remplacer la file d'attente de transmission par défaut de chaque canal émetteur de cluster par `SYSTEM.CLUSTER.TRANSMIT.ChannelName` en modifiant l'attribut de gestionnaire de files d'attente `DEFCLXQ`. Vous ne pouvez pas supprimer `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Il est également utilisé pour définir les vérifications d'autorisation si la file d'attente de transmission par défaut utilisée est `SYSTEM.CLUSTER.TRANSMIT.QUEUE` ou `SYSTEM.CLUSTER.TRANSMIT.ChannelName`.

### **SYSTEM.DEF.CLUSRCVR**

Chaque cluster dispose d'une définition de canal `CLUSRCVR` par défaut appelée `SYSTEM.DEF.CLUSRCVR`. `SYSTEM.DEF.CLUSRCVR` permet de fournir les valeurs par défaut pour les attributs que vous ne spécifiez pas lorsque vous créez un canal récepteur de cluster sur un gestionnaire de files d'attente dans le cluster.

### **SYSTEM.DEF.CLUSSDR**

Chaque cluster dispose d'une définition de canal `CLUSSDR` par défaut appelée `SYSTEM.DEF.CLUSSDR`. `SYSTEM.DEF.CLUSSDR` permet de fournir les valeurs par défaut pour les attributs que vous ne spécifiez pas lorsque vous créez un canal émetteur de cluster sur un gestionnaire de files d'attente dans le cluster.

## Concepts associés

Utilisation des objets de cluster par défaut

# Messagerie de type publication/abonnement

La messagerie de type publication/abonnement vous permet de dissocier le fournisseur d'informations des consommateurs de ces informations. Il n'est pas nécessaire que l'application émettrice et l'application de réception se connaissent pour que les informations puissent être envoyées et reçues.

Pour qu'une application IBM MQ point-à-point puisse envoyer un message à une autre application, elle doit connaître cette application. Par exemple, elle doit connaître le nom de la file d'attente vers laquelle envoyer les informations et peut également indiquer un nom de gestionnaire de files d'attente.

Avec IBM MQ, votre application n'a pas besoin de connaître l'application cible. Toutes les application émettrices doivent exécuter les opérations suivantes :

- Insérer un message IBM MQ qui contient les informations nécessaire à l'application.
- Affecter le message à une rubrique indiquant le sujet des informations.
- Laisser IBM MQ traiter la distribution de ces informations.

De façon similaire, l'application cible peut ignorer la source des informations qu'elle reçoit.

La figure suivante illustre le système de publication/abonnement le plus simple. Il est composé d'un diffuseur de publications, d'un gestionnaire de files d'attente et d'un abonné. Un abonnement est effectué par l'abonné auprès du gestionnaire de files d'attente, une publication est envoyée depuis le diffuseur de publications vers le gestionnaire de files d'attente, puis est transmise par le gestionnaire de files d'attente à l'abonné.

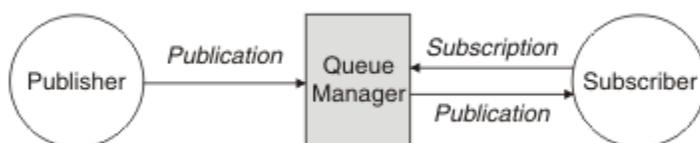


Figure 17. Configuration de publication/abonnement simple

Un système de publication/abonnement standard est composé de plusieurs diffuseurs de publications et d'abonnés à de nombreuses rubriques différentes, et souvent, de plusieurs gestionnaires de files d'attente. Une application peut être à la fois un éditeur et un abonné.

Une autre différence importante entre la messagerie de publication/abonnement et le mode point-à-point est qu'un message envoyé à une file d'attente point-à-point est traité uniquement par une seule application concernée. Un message publié dans une rubrique de publication/abonnement, pour laquelle plusieurs abonnés ont manifesté un intérêt, est traité par chaque abonné intéressé.

## Composants de publication/abonnement

La fonction de publication/abonnement est le mécanisme par lequel les abonnés peuvent recevoir des informations de diffuseurs de publications sous la forme de messages. Les interactions entre les diffuseurs et les abonnés sont contrôlées par des gestionnaires de files d'attente à l'aide de fonctions IBM MQ standard.

Un système de publication/abonnement standard est composé de plusieurs diffuseurs de publications et d'abonnés à de nombreuses rubriques différentes, et souvent, de plusieurs gestionnaires de files d'attente. Une application peut être à la fois un éditeur et un abonné.

Le fournisseur des informations est appelé le *diffuseur de publications*. Les diffuseurs de publications fournissent des informations sur un sujet, sans avoir besoin de connaître les applications qu'elles intéressent. Ils génèrent ces informations sous la forme de messages, appelés *publications*, qu'ils souhaitent publier et définissent la rubrique de ces messages.

Le consommateur de ces informations est appelé un *abonné*. Les abonnés créent des *abonnements* qui décrivent la rubrique qui intéresse l'abonné. L'abonnement détermine donc les publications transmises à l'abonné. Les abonnés peuvent souscrire différents abonnements et recevoir des informations de différents diffuseurs de publications.

Les informations publiées sont envoyées dans un message IBM MQ, et leur sujet est identifié par sa *rubrique*. Lorsqu'il publie des informations, le diffuseur de publications les associe à une rubrique et l'abonné indique les rubriques à propos desquelles il souhaite recevoir des publications. L'abonné reçoit des informations concernant uniquement les rubriques auxquelles il s'est abonné.

C'est l'existence de rubriques qui permet aux fournisseurs et consommateurs d'informations d'être dissociés de la messagerie de type publication/abonnement, ce qui élimine le besoin d'inclure une destination spécifique dans chaque message, comme dans la messagerie point à point.

Les interactions entre les diffuseurs et les abonnés sont contrôlées par un gestionnaire de files d'attente. Le gestionnaire de files d'attente reçoit les messages envoyés par les diffuseurs de publication et les abonnements des abonnés (à un certain nombre de rubriques). Le travail du gestionnaire de files d'attente consiste à acheminer les messages publiés vers les abonnés susceptibles d'être intéressés par la rubrique des messages.

Des fonctions standard IBM MQ sont utilisées pour distribuer les messages afin que les applications puissent tirer parti de toutes les fonctionnalités disponibles pour les applications IBM MQ existantes. Cela signifie que vous pouvez utiliser des messages persistants pour bénéficier d'une distribution assurée une seule fois et que vos messages peuvent faire partie d'une unité d'oeuvre transactionnelle afin de garantir qu'ils sont distribués à l'abonné uniquement s'ils sont validés par le diffuseur de publications.

## **Diffuseurs de publications et publications**

Dans la publication/l'abonnement IBM MQ, un diffuseur est une applications qui fournit des informations sur une rubrique à un gestionnaire de files d'attente sous la forme d'un message standard IBM MQ appelé publication. Un diffuseur de publications peut publier des informations sur plusieurs rubriques.

Les diffuseurs de publications utilisent l'instruction MQPUT pour placer un message (publication) dans une rubrique précédemment ouverte. Le gestionnaire de files d'attente local achemine ensuite la publication vers les abonnés qui possèdent un abonnement à la rubrique de la publication. Un message publié peut être utilisé par plusieurs abonnés.

Outre la distribution des publications à l'ensemble des abonnés locaux qui possèdent des abonnements appropriés, un gestionnaire de files d'attente peut également distribuer la publication à d'autres gestionnaires de files d'attente connectés, soit directement ou via un réseau de gestionnaires de files d'attente possédant des abonnements à la rubrique.

Dans un réseau de publication/abonnement IBM MQ, une application de publication peut être également un abonné.

## **Publications sous le contrôle d'un point de synchronisation**

Les diffuseurs de publications peuvent émettre des appels MQPUT ou MQPUT1 sous un point de synchronisation afin d'inclure tous les messages distribués aux abonnés dans une unité d'oeuvre. Si l'option MQPMO\_RETAIN, ou les options de distribution de rubrique NPMMSGDLV ou PMSGDLV avec les valeurs ALL ou ALLDUR, sont spécifiées, le gestionnaire de files d'attente utilise des appels MQPUT ou MQPUT1 internes dans un point de synchronisation, dans la portée du diffuseur de publications MQPUT ou de l'appel MQPUT1.

## **Informations d'état et données sur l'événement**

Les publications peuvent être catégorisées en publications d'état, comme le cours actuel d'une action, ou en publications d'événement, comme la vente de cette action.

## **Publications relatives à l'état**

Les *publications d'état* contiennent des informations sur l'état actuel d'un élément, comme le cours d'une action ou le score actuel d'un match de football. Lorsqu'un événement se produit, (par exemple, le prix des articles est modifié ou le score d'un match change), les informations relatives à l'état précédent sont obsolètes et sont remplacées par les informations les plus récentes.

Un abonné souhaite recevoir la version actuelle des informations d'état au démarrage, puis recevoir les nouvelles informations chaque fois que l'état change.

Si une publication contient des informations d'état, elle est souvent publiée en tant que publication conservée. Un nouvel abonné souhaite généralement obtenir immédiatement les informations d'état en cours sans attendre qu'un événement entraîne la une nouvelle publication des informations. Les abonnés recevront automatiquement la publication conservée d'une rubrique lorsqu'ils s'abonnent sauf s'ils utilisent les options MQSO\_PUBLICATIONS\_ON\_REQUEST ou MQSO\_NEW\_PUBLICATIONS\_ONLY.

## **Publications relatives aux événements**

Les *publications d'événement* contiennent des informations sur des événements individuels qui se produisent, comme la vente d'une action ou le marquage d'un but. Chaque événement est indépendant des autres événements.

Un abonné souhaite recevoir des informations sur les événements au fur et à mesure qu'ils se produisent.

### **Publications conservées**

Par défaut, après qu'une publication a été envoyée à tous les abonnés intéressés, elle est supprimée. Toutefois, un diffuseur de publications peut indiquer qu'une copie d'une publication est conservée pour être envoyée aux abonnés ultérieurs qui manifestent un intérêt pour cette rubrique.

La suppression de publications après leur envoi à tous les abonnés intéressés est adaptée aux informations d'événement, mais pas toujours aux informations d'état. Lorsqu'un message est conservé, les nouveaux abonnés ne doivent pas attendre que les informations d'état initial soient à nouveau publiées pour les recevoir. Par exemple, un abonné souscrivant un abonnement au cours d'une action recevra immédiatement le cours actuel de l'action, sans attendre que celui-ci change (et soit donc publié à nouveau).

Le gestionnaire de files d'attente ne peut conserver qu'une publication par rubrique pour que la publication conservée existante d'une rubrique soit supprimée dès qu'il reçoit une nouvelle publication conservée. Toutefois, la suppression de la publication existante peut ne pas avoir lieu de façon synchrone avec l'arrivée de la nouvelle publication conservée. Par conséquent, dans la mesure du possible, faites en sorte qu'un seul diffuseur de publications envoie les publications conservées d'une rubrique.

Les abonnés peuvent préciser qu'ils ne souhaitent pas recevoir de publications conservées en utilisant l'option d'abonnement MQSO\_NEW\_PUBLICATIONS\_ONLY. Les abonnés existants peuvent demander à recevoir des copies en double des publications conservées.

Vous pouvez parfois ne pas souhaiter conserver des publications, même pour les informations d'état :

- Si tous les abonnements à une rubrique sont souscrits avant que des publications soient effectuées sur cette rubrique et que vous ne prévoyez pas ou n'autorisez pas de nouveaux abonnements, il n'est pas nécessaire de conserver les publications car celles-ci sont distribuées à l'ensemble des abonnés la première fois qu'elles sont publiées.
- Si des publications ont lieu fréquemment, par exemple, toutes les secondes, un nouvel abonné (ou un abonné en reprise après un échec) reçoit l'état actuel presque immédiatement après son abonnement initial et il n'est donc pas nécessaire de conserver ces publications.
- Si les publications sont volumineuses, vous risquez d'avoir besoin d'un espace de stockage considérable pour stocker les publications conservées pour chaque rubrique. Dans un environnement à plusieurs gestionnaires de files les publications conservées sont stockées par tous les gestionnaires de files d'attente du réseau qui ont un abonnement correspondant.

Lorsque vous décidez d'utiliser ou non les publications conservées, tenez compte de la manière dont les applications d'abonnement sont récupérées après un incident. Si le diffuseur de publications n'utilise pas de publications conservées, l'application d'abonné peut avoir besoin de stocker son état actuel en local.

Pour qu'une publication soit conservée, utilisez l'option d'insertion de message MQPMO\_RETAIN. Si cette option est utilisée et que la publication ne peut pas être conservée, le message n'est pas publié et l'appel échoue avec MQRC\_PUT\_NOT\_RETAINED.

Si un message est une publication conservée, cela est indiquée par la propriété de message MQIsRetained. La persistance d'un message est telle que lorsque celui-ci a été publié initialement.

### Concepts associés

Remarques de conception pour les publications conservées dans les clusters de publication/abonnement

### **Publications sous le contrôle d'un point de synchronisation**

Dans la fonction de publication/abonnement IBM MQ, le point de synchronisation peut être utilisé par des diffuseurs ou en interne par le gestionnaire de files d'attente.

Les diffuseurs utilisent le point de synchronisation lorsqu'ils émettent des appels MQPUT/MQPUT1 avec l'option MQPMO\_SYNCPOINT. Tous les messages transmis aux abonnés font partie du nombre maximal de messages dans une unité d'oeuvre. L'attribut du gestionnaire de files d'attente MAXUMSGS spécifie cette limite. Si la limite est atteinte, l'éditeur reçoit le code anomalie 2024 (07E8) (RC2024): MQRC\_SYNCPOINT\_LIMIT\_REACHED.

Lorsqu'un diffuseur émet des appels MQPUT/MQPUT1 à l'aide de MQPMO\_NO\_SYNCPOINT avec l'option MQPMO\_RETAIN ou les options de distribution de rubrique NPMSGDLV/PMSGDLV avec des valeurs ALL ou ALLDUR, le gestionnaire de files d'attente utilise des points de synchronisation internes pour s'assurer que les messages sont transmis comme il convient. Le diffuseur de publications peut recevoir le code anomalie 2024 (07E8) (RC2024): MQRC\_SYNCPOINT\_LIMIT\_REACHED si la limite est atteinte dans la portée de l'appel MQPUT/MQPUT1 de l'éditeur.

### Abonnés et abonnements

Dans la publication/abonnement d'IBM MQ, un abonné est une application qui demande des informations sur une rubrique spécifique à un gestionnaire de files d'attente dans un réseau de publication/abonnement. Un abonné peut recevoir des messages sur la même rubrique ou des rubriques différentes, depuis plusieurs diffuseur de publications.

Des abonnements peuvent être créés manuellement via une commande MQSC ou par des applications. Ces abonnements sont envoyés au gestionnaire de files d'attente local et contiennent des informations sur les publications que l'abonné souhaite recevoir :

- La rubrique qui intéresse l'abonné ; il peut s'agir de plusieurs rubriques si des caractères génériques sont utilisés.
- Une chaîne de sélection facultative à appliquer aux messages publiés.
- Un descripteur vers une file d'attente (appelée *file d'attente de souscription*), dans laquelle les publications sélectionnées doivent être placées, et l'ID de corrélation (CorrelId) facultatif.

Le gestionnaire de files d'attente local stocke les informations d'abonnement. Lorsqu'il reçoit une publication, il analyse ces informations pour déterminer si un abonnement correspond à la rubrique et la chaîne de sélection de la publication. Pour chaque abonnement qui correspond, le gestionnaire de files d'attente dirige la publication vers la file d'attente de souscription de l'abonné. Les informations sur les abonnements stockées par gestionnaire de files d'attente peuvent être affichées à l'aide des commandes DIS SUB et DIS SBSTATUS.

Un abonnement est supprimé uniquement lorsque l'un des événements suivants se produit :

- L'abonné se désabonne à l'aide de l'appel MQCLOSE (si l'abonnement a été défini comme non durable).
- L'abonnement expire.
- L'abonnement est supprimé par l'administrateur système à l'aide de la commande DELETE SUB.
- L'application d'abonné s'arrête (si l'abonnement a été défini comme non durable).

- Le gestionnaire de files d'attente est arrêté ou redémarré (si l'abonnement a été défini comme non durable).

Lors de l'extraction des messages, utilisez les options appropriées dans l'appel MQGET. Si votre application traite uniquement les messages d'un abonnement, vous devez utiliser au moins `get-by-correlid`, comme illustré dans l'exemple de programme C `amqssbxa.c` et dans l'exemple `Unmanaged MQ subscriber`. Le **CorrelId** à utiliser est renvoyé par MQSUB dans le MQSD.Zone **SubCorrelId**.

### Concepts associés

[Abonnements clonés et partagés](#)

### Référence associée

[Exemples de définition de la propriété sharedSubscription](#)

## Files d'attente gérées et publication/abonnement

Lorsque vous créez un abonnement vous pouvez choisir d'utiliser la mise en file d'attente gérée. Si vous utilisez une mise en file d'attente gérée, une file d'attente de souscription est créée automatiquement lorsque vous créez un abonnement. Les files d'attente gérées sont nettoyées automatiquement en fonction de la durabilité de l'abonnement. Avec l'utilisation de files d'attente gérées, vous n'avez pas besoin de créer des files d'attente pour recevoir des publications et les publications inutilisées sont supprimées automatiquement des files d'attente de souscription si une connexion d'abonnement non durable est fermée.

Si une application n'a pas besoin d'utiliser une file d'attente particulière comme file d'attente de souscription (destination des publications qu'elle reçoit), elle peut utiliser des *abonnements gérés* à l'aide de l'option d'abonnement MQSO\_MANAGED. Si vous créez un abonnement géré, le gestionnaire de files d'attente renvoie un identificateur d'objet à l'abonné pour une file d'attente de souscription créée par le gestionnaire de files d'attente, dans laquelle les publications seront reçues. Cela est dû au fait qu'un *abonnement géré* a pour gestionnaire IBM MQ. L'identificateur d'objet de la file d'attente sera renvoyé pour vous permettre de rechercher, obtenir ou interroger la file d'attente. (Il n'est pas possible d'insérer des éléments dans une file d'attente gérée ou de définir des attributs pour celle-ci, sauf si on vous a accordé explicitement l'accès aux files d'attente dynamiques temporaires.)

La durabilité de l'abonnement détermine si la file d'attente gérée est conservée lorsque la connexion de l'application abonnée au gestionnaire de files d'attente est interrompue.

Les abonnements gérés sont particulièrement utiles dans le cas d'abonnements non durables, car sinon, lorsque la connexion de l'application est arrêtée, les messages inutilisés restent dans la file d'attente de l'abonné en y occupant indéfiniment de l'espace. Si vous utilisez un abonnement géré, la file d'attente gérée est file d'attente dynamique temporaire et est donc supprimée avec tous les messages inutilisés lorsque la connexion est interrompue pour l'une des raisons suivantes :

- MQCLOSE avec MQCO\_REMOVE\_SUB est utilisé et l'identificateur d'objet Hobj géré est fermé.
- Une connexion vers une application utilisant un abonnement non durable (MQSO\_NON\_DURABLE) est perdue.
- Un abonnement est supprimé car il est arrivé à expiration et l'identificateur Hobj géré est fermé.

Des abonnements gérés peuvent également être utilisés avec des abonnements durables mais il est possible que vous souhaitiez laisser les messages inutilisés dans la file d'attente de souscription pour pouvoir les récupérer lorsque la connexion est rouverte. C'est pourquoi les files d'attente gérées pour les abonnements durables prennent la forme d'une file d'attente dynamique permanente qui est conservée lorsque la connexion de l'application abonnée au gestionnaire de files d'attente est interrompue.

Vous pouvez définir un délai d'expiration pour votre abonnement si vous souhaitez utiliser une file d'attente gérée dynamique permanente qui, si elle sera conservée après l'interruption de la connexion, ne continuera pas d'exister indéfiniment.

Si vous supprimez la file d'attente gérée, vous recevez un message d'erreur.

Les files d'attente gérées qui sont créées sont nommées avec des nombres à la fin de leur nom (des horodatages) pour être uniques.

## **Durabilité des abonnements**

Les abonnements peuvent être configurés pour être non durables ou durables. La durabilité des abonnements détermine ce qui se produit pour des abonnements lorsque des applications d'abonnement se déconnectent d'un gestionnaire de files d'attente.

### **Abonnements durables**

Les abonnements durables continuent d'exister lorsque la connexion d'une application d'abonnement au gestionnaire de files d'attente est fermée. Si un abonnement est durable, lorsque l'application d'abonnement se déconnecte, l'abonnement est conservé et il peut être utilisé par l'application d'abonnement lorsque celle-ci se reconnecte et redemande l'abonnement en utilisant le même nom **SubName** que celui renvoyé lors de la création de l'abonnement.

Lors d'une souscription à un abonnement durable, un nom d'abonnement (**SubName**) est obligatoire. Les noms d'abonnement doivent être uniques dans un gestionnaire de files d'attente pour permettre d'identifier les abonnements. Cette méthode d'identification est nécessaire lorsque vous spécifiez un abonnement que vous souhaitez reprendre si vous avez volontairement fermé la connexion à l'abonnement (à l'aide de l'option MQCO\_KEEP\_SUB) ou si vous avez été déconnecté du gestionnaire de files d'attente. Vous pouvez reprendre un abonnement existant en utilisant l'appel MQSUB avec l'option MQSO\_RESUME. Les noms d'abonnement sont également affichés si vous utilisez la commande DISPLAY SBSTATUS avec SUBTYPE ALL ou ADMIN.

Lorsqu'une application n'a plus besoin d'un abonnement durable, celui-ci peut être supprimé à l'aide de l'appel de fonction MQCLOSE avec l'option MQCO\_REMOVE\_SUB ou supprimé manuellement à l'aide de la commande MQSC DELETE SUB.

Vous pouvez utiliser l'attribut de rubrique **DURSUB** pour indiquer si des abonnements durables peuvent ou non être souscrits sur une rubrique.

Renvoyée à partir d'un appel MQSUB à l'aide de l'option MQSO\_RESUME, le délai d'expiration d'abonnement est défini sur le délai d'expiration d'origine de l'abonnement et non sur le délai d'expiration restant.

Un gestionnaire de files d'attente continue d'envoyer des publications pour répondre à un abonnement durable même si l'application d'abonnement correspondante n'est pas connectée. Cela entraîne une accumulation de messages dans la file d'attente de l'abonné. Le moyen le plus simple pour éviter ce problème est d'utiliser un abonnement non durable chaque fois que cela convient. Cependant, lorsqu'il est nécessaire d'utiliser des abonnements durables, l'accumulation des messages peut être évitée si l'abonné souscrit à un abonnement à l'aide de l'option Publications conservées. Un abonné peut alors contrôler la réception des publications à l'aide de l'appel MQSUBRQ.

### **Abonnements non durables**

Les abonnements non durables existent tant que la connexion entre l'application d'abonnement et le gestionnaire de files d'attente est ouverte. L'abonnement est supprimé lorsque l'application d'abonnement se déconnecte du gestionnaire de files d'attente volontairement ou suite à une perte de connexion. Lorsque la connexion est fermée, les informations concernant l'abonnement sont supprimées du gestionnaire de files d'attente, et n'apparaissent plus si vous affichez les abonnements à l'aide de la commande DISPLAY SBSTATUS. Aucun autre message n'est placé dans la file d'attente de souscription.

Ce qui se produit pour les publications non consommées dans une file d'attente de souscription pour des abonnements non durables est déterminé comme suit.

- Si une application d'abonnement utilise une destination gérée, toutes les publications qui n'ont pas été consommées sont automatiquement supprimées.
- Si l'application d'abonnement fournit un identificateur à sa propre file d'attente de souscription lorsqu'il s'abonne, les messages non consommés ne sont pas supprimés automatiquement. Il incombe à l'application de vider la file d'attente si c'est pertinent. Si la file d'attente est partagée par plusieurs abonnés ou d'autres applications point-à-point, cela peut ne pas être approprié de la vider complètement.

Bien que cela ne soit pas obligatoire pour les abonnements non durables, un nom d'abonnement, s'il est fourni, est utilisé par le gestionnaire de files d'attente. Les noms d'abonnement doivent être uniques dans le gestionnaire de files d'attente pour permettre d'identifier les abonnements.

### **Concepts associés**

[Abonnements clonés et partagés](#)

### **Tâches associées**

[Utilisation d'abonnements partagés JMS 2.0](#)

### **Référence associée**

[Exemples de définition de la propriété `sharedSubscription`](#)

## **Chaînes de sélection**

Une *chaîne de sélection* est une expression qui est appliquée à une publication pour déterminer si elle correspond à un abonnement. Les chaînes de sélection peuvent inclure des caractères génériques.

Lorsque vous vous abonnez, en plus d'une rubrique, vous pouvez spécifier une chaîne de sélection pour sélectionner les publications en fonction de leurs propriétés de message.

La chaîne de sélection est évaluée par rapport au message inséré par le diffuseur de publications avant d'être modifiée pour la distribution à chaque abonné. Faites attention lorsque vous utilisez des zones dans la chaîne de sélection, susceptibles d'être modifiées dans le cadre de l'opération de publication. Par exemple, les zones `MQMD.UserIdentifier`, `MsgId` et `CorrelId`.

Les chaînes de sélection ne doivent pas faire référence à des zones de propriété de message ajoutées par le gestionnaire de files d'attente dans le cadre de l'opération de publication (voir [Propriétés des messages de publication/abonnement](#)), à l'exception de la propriété de message `MQTopicString`, qui contient la chaîne de rubrique pour la publication.

### **Concepts associés**

[Règles et restrictions des chaînes de sélection](#)

## **Rubriques**

Une rubrique est le sujet d'une information publiée dans un message de publication/abonnement.

Les messages dans les systèmes point-à-point sont envoyés à une adresse de destination spécifique. Les messages se trouvant dans des systèmes basés sur des sujets de publication/abonnement sont envoyés à des abonnés en fonction des sujets décrivant le contenu du message. Dans des systèmes basés sur le contenu, les messages sont envoyés aux abonnés en fonction du contenu du message.

Le système de publication/abonnement IBM MQ est un système de publication/abonnement basé sur les sujets. Un diffuseur de publication crée un message et le publie avec une chaîne de rubrique qui convient le mieux au sujet de la publication. Pour recevoir des publications, un abonné crée un abonnement avec un schéma correspondant à la chaîne de rubrique pour sélectionner les rubriques de publication. Le gestionnaire de files d'attente distribue les publications aux abonnés dont les abonnements correspondent à la rubrique de publication, et qui sont autorisés à recevoir les publications. L'article, «[Chaînes de rubrique](#)», à la [page 73](#), décrit la syntaxe des chaînes de rubrique qui identifient le sujet d'une publication. Les abonnés créent également des chaînes de rubriques pour sélectionner les rubriques qu'ils souhaitent recevoir. Les chaînes de rubrique que les abonnés créent peuvent contenir un des deux schémas de caractères génériques alternatifs pour correspondre aux chaînes de rubrique des publications. La correspondance des schémas est décrite dans «[Schémas de caractères génériques](#)», à la [page 74](#).

Dans une publication/un abonnement basé sur le sujet, les diffuseurs de publications ou les administrateurs, ont pour tâche de classer les sujets dans les rubriques. Généralement, les sujets sont classés dans un ordre hiérarchique, dans des arborescences de rubriques, à l'aide du caractère `' / '` pour créer de sous-rubriques dans la chaîne de rubrique. Voir «[Arborescence de rubriques](#)», à la [page 80](#) pour des exemples d'arborescences de rubriques. Les rubriques sont des noeuds de l'arborescence. Les rubriques sont des noeuds qui ne comportent aucune sous-rubrique, ou des noeuds intermédiaires avec des sous-rubriques.

Parallèlement à l'organisation de rubriques en arborescence hiérarchique de rubriques, vous pouvez associer des rubriques à des objets de rubriques d'administration. Vous affectez des attributs à une rubrique, par exemple un attribut indiquant si la rubrique est distribuée dans un cluster, en l'associant à un objet de rubrique d'administration. L'association est faite en nommant la rubrique à l'aide de l'attribut TOPICSTR de l'objet de rubrique d'administration. Si vous n'associez pas explicitement un objet de rubrique d'administration à une rubrique, elle hérite des attributs de son ancêtre le plus proche dans l'arborescence de rubriques que vous avez associé à un objet de rubrique d'administration. Si vous n'avez pas défini de rubrique parent, elle hérite de SYSTEM.BASE.TOPIC. Les objets de rubrique d'administration sont décrits dans «Objets de rubrique d'administration», à la page 81.

**Remarque :** Même si vous héritez de tous les attributs d'une rubrique de SYSTEM.BASE.TOPIC, définissez une rubrique principale pour vos rubriques qui hérite directement de SYSTEM.BASE.TOPIC. Par exemple, dans l'espace de rubrique des États américains, USA/Alabama USA/Alaska, et ainsi de suite, USA est le sujet principal. L'objectif premier de la rubrique principale est de créer des espaces de sujet discrets et sans chevauchements afin d'éviter que des publications soient associées aux mauvais abonnements. Cela signifie également que vous pouvez modifier les attributs de votre rubrique principale pour affecter l'espace de sujet dans son intégralité. Par exemple, vous pouvez définir le nom de l'attribut **CLUSTER**.

Lorsque vous faites référence à une rubrique en tant que diffuseur de publications ou en tant qu'abonné, vous avez le choix entre fournir une chaîne de rubrique ou faire référence à un objet de rubrique. Vous pouvez également faire les deux, auquel cas la chaîne de rubrique que vous avez fournie définit une sous-rubrique de l'objet de rubrique. Le gestionnaire de files d'attente identifie la rubrique en ajoutant la chaîne de rubrique au préfixe de la chaîne de rubrique nommé dans l'objet de rubrique, en insérant un '/' additionnel entre les deux chaînes de rubriques, par exemple, *chaîne de rubrique/chaîne d'objet*. «Combinaison de chaînes de rubrique», à la page 78 décrit cela en plus de détails. La chaîne de rubrique résultante est utilisée pour identifier la rubrique et l'associer à un objet de rubrique d'administration. L'objet de rubrique d'administration n'est pas nécessairement le même objet de rubrique que l'objet de rubrique correspondant à la rubrique principale.

Dans une publication/un abonnement basé sur le contenu, vous définissez les messages que vous souhaitez recevoir en fournissant des chaînes de sélection recherchant le contenu de chaque message. IBM MQ fournit une forme intermédiaire de publication/abonnement basée sur le contenu à l'aide des sélecteurs de message qui analysent les propriétés du message plutôt que le contenu complet du message. Voir *Sélecteurs*. L'utilité archétype des sélecteurs de messages est de s'abonner à une rubrique et ensuite de qualifier la sélection avec une propriété numérique. Le sélecteur vous permet d'indiquer les valeurs qui vous intéressent uniquement dans une certaine plage ; quelque chose que vous ne pouvez pas faire en utilisant un caractère ou des caractères génériques basés sur des rubriques. Si vous devez filtrer en vous basant sur le contenu complet du message, vous devez utiliser IBM Integration Bus.

### **Chaînes de rubrique**

Le libellé de la rubrique est publiée à l'aide d'une chaîne de rubrique. Abonnez-vous aux groupes de rubriques en utilisant des chaînes de caractères génériques de type caractère ou de type rubrique.

### **Rubriques**

Une *chaîne de rubrique* est une chaîne de caractères qui identifie la rubrique d'un message de publication/abonnement. Vous pouvez utiliser n'importe quels caractères lorsque vous construisez une chaîne de rubrique.



Trois caractères ont une signification spéciale dans la fonction de publication/abonnement de la IBM WebSphere MQ 7. Ils sont autorisés dans une chaîne de rubrique, mais utilisez-les avec précaution. L'utilisation des caractères spéciaux est expliquée dans «Schéma de caractères génériques basés sur des rubriques», à la page 75.

## Barre oblique (/)

Séparateur de niveaux de rubrique. Utilisez le caractère '/' pour structurer la rubrique dans une arborescence de rubriques.

Évitez les niveaux de rubrique vides, '// ' dans la mesure du possible. Ils correspondent à des noeuds de la hiérarchie de rubriques sans chaîne de rubrique. Une barre oblique '/' de début ou de fin dans une chaîne de rubrique correspond à un noeud vide de début ou de fin et doit être aussi évitée.

## Signe dièse (#)

Combiné avec '/' pour construire un caractère générique multi-niveau dans des abonnements. Veillez à utiliser '#' à côté de '/' dans des chaînes de rubrique servant à nommer des rubriques publiées. «Exemples de chaînes de rubrique», à la page 74 montre une utilisation appropriée de '# '.

Les chaînes '.../#/...', '#/...' et '.../#' ont une signification particulière dans les chaînes de rubrique d'abonnement. Elles correspondent à toutes les rubriques à un ou plusieurs niveaux dans la hiérarchie de rubriques. Par conséquent, si vous avez créé une rubrique avec l'une de ces séquences, vous ne pouvez pas vous y abonner sans aussi vous abonner à toutes les rubriques à plusieurs niveaux de la hiérarchie de rubriques.

## Signe plus (+)

Combiné avec '/' pour construire un caractère générique à un niveau dans des abonnements. Veillez à utiliser '+' à côté de '/' dans des chaînes de rubrique servant à nommer des rubriques publiées.

Les chaînes '.../+/...', '+/...' et '.../+' ont une signification particulière dans les chaînes de rubrique d'abonnement. Elles correspondent à toutes les rubriques à un niveau dans la hiérarchie de rubriques. Par conséquent, si vous avez créé une rubrique avec l'une de ces séquences, vous ne pouvez pas vous y abonner sans aussi vous abonner à toutes les rubriques à un niveau de la hiérarchie de rubriques.

## Exemples de chaînes de rubrique

```
IBM/Business Area#/Results
IBM/Diversity/%African American
```

## Référence associée

[TOPIC](#)

### *Schémas de caractères génériques*

Deux schémas de caractères génériques sont utilisés pour s'abonner à plusieurs rubriques. Le choix du schéma est une option d'abonnement.

### **MQSO\_WILDCARD\_TOPIC**

Sélection des rubriques auxquelles s'abonner à l'aide du schéma de caractères génériques basés sur les rubriques.

Il s'agit du schéma par défaut si aucun schéma générique n'est sélectionné par défaut.

### **MQSO\_WILDCARD\_CHAR**

Sélection des rubriques auxquelles s'abonner à l'aide du schéma de caractères génériques de type caractère.

Définissez l'un des schémas en spécifiant le paramètre **wschema** sur la commande DEFINE SUB. Pour plus d'informations, voir [DEFINE SUB](#).

**Remarque :** Les abonnements créés avant IBM WebSphere MQ 7.0 utilisent toujours le schéma de caractères génériques.

## Exemples

```
IBM/+/Results
```

```
#/Results
IBM/Software/Results
IBM/*ware/Results
```

### *Schéma de caractères génériques basés sur des rubriques*

Les caractères génériques basés sur des rubriques permettent de s'abonner à plusieurs rubriques en même temps.

Les caractères génériques basés sur les rubriques représentent une fonction puissante du système de rubriques dans la publication/l'abonnement IBM MQ. Les caractères génériques multi-niveau et à un niveau peuvent être utilisés pour les abonnements, mais ils ne peuvent pas l'être par le diffuseur d'un message dans une rubrique.

Le schéma de caractères génériques basés sur des rubriques vous permet de sélectionner des publications regroupées par niveau de rubrique. Vous pouvez décider, pour chaque niveau de la hiérarchie de rubriques, si la chaîne dans l'abonnement pour ce niveau de rubrique doit correspondre exactement à la chaîne dans la publication ou pas. Par exemple, l'abonnement IBM/+/Results sélectionne toutes les rubriques.

```
IBM/Software/Results
IBM/Services/Results
IBM/Hardware/Results
```

Il existe deux types de caractères génériques.

#### **Caractère générique multi-niveau**

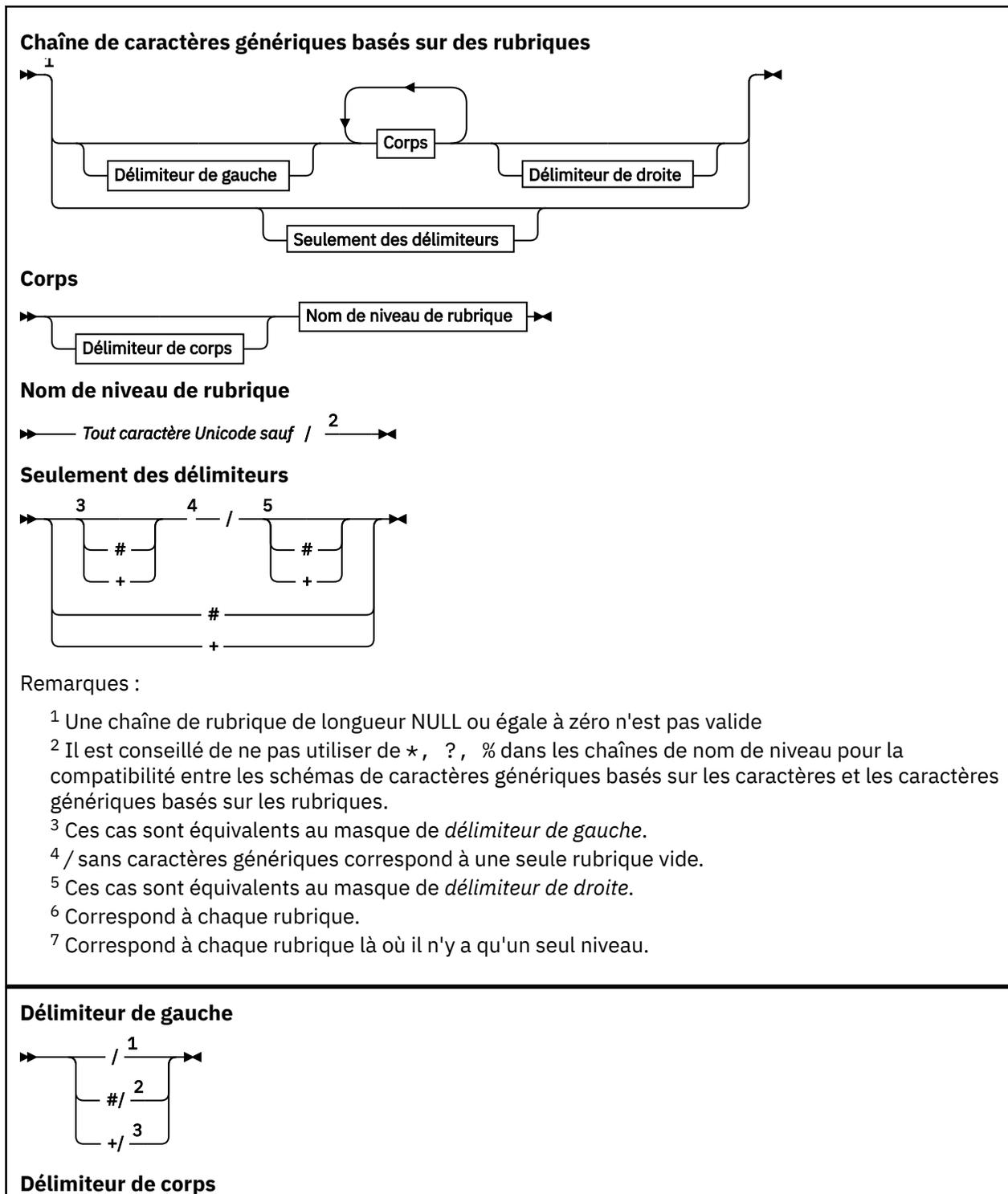
- Le caractère générique multi-niveau est utilisé dans des abonnements. Lorsqu'il est utilisé dans une publication, il est traité comme un littéral.
- Le caractère générique multi-niveau '# ' est utilisé pour correspondre aux divers niveaux d'une rubrique. Par exemple, dans l'exemple d'arborescence de rubriques, si vous vous abonnez à 'USA/Alaska/# ', vous recevez des messages sur les rubriques 'USA/Alaska ' et 'USA/Alaska/Juneau '.
- Le caractère générique multi-niveau peut représenter zéro, un ou plusieurs niveaux. Par conséquent, 'USA/# ' peut également correspondre à 'USA ', '# ' ne représentant alors aucun niveau. Le séparateur de niveaux de rubrique ne sert à rien dans ce contexte, car il n'y a aucun niveau à séparer.
- Le caractère générique multi-niveau ne s'applique que s'il est spécifié seul ou après le caractère séparateur de niveaux de rubrique. Par conséquent, '# ' et 'USA/# ' sont des rubriques valides, dans lesquelles le caractère '# ' est traité comme un caractère générique. Toutefois, même si 'USA# ' est également une chaîne de rubrique valide, le caractère '# ' n'est pas considéré comme un caractère générique et ne possède pas de signification spéciale. Pour plus d'informations, voir [«Cas où les caractères génériques basés sur des rubriques ne sont pas génériques»](#), à la page 77.

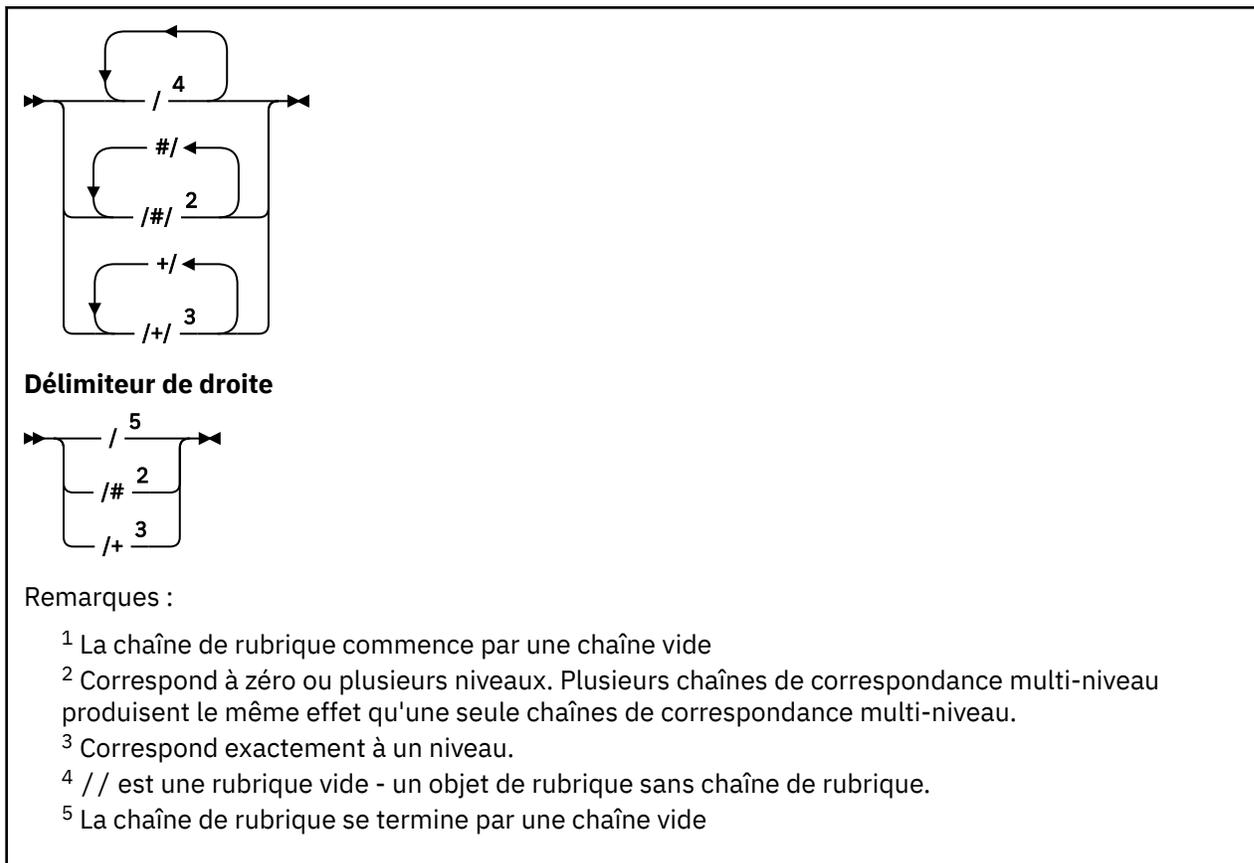
#### **Caractère générique à un niveau**

- Le caractère générique à un niveau est utilisé dans les abonnements. Lorsqu'il est utilisé dans une publication, il est traité comme un littéral.
- Le caractère générique à un niveau '+ ' ne correspond qu'à un seul niveau de rubrique. Par exemple, 'USA/+ ' correspond à 'USA/Alabama ', mais pas à 'USA/Alabama/Auburn '. Le caractère générique à un niveau ne correspondant qu'à un seul niveau, 'USA/+ ' ne correspond pas à 'USA '.
- Le caractère générique à un niveau peut être utilisé à tout niveau de l'arborescence de rubriques et avec le caractère générique multi-niveau. Le caractère générique à un niveau doit être spécifié après le séparateur de niveaux de rubrique, sauf s'il est spécifié seul. Par conséquent, '+ ' et 'USA/+ ' sont des rubriques valides, dans lesquelles le caractère '+ ' est traité comme un caractère générique. Toutefois, même si 'USA+ ' est également une chaîne de rubrique valide, le caractère '+ ' n'est pas considéré comme un caractère générique et ne possède pas de signification spéciale. Pour plus d'informations, voir [«Cas où les caractères génériques basés sur des rubriques ne sont pas génériques»](#), à la page 77.

La syntaxe pour le schéma de caractères génériques basés sur des rubriques ne comporte pas de caractères d'échappement. Le fait que '#' et '+' soient traités ou non comme des caractères génériques dépend de leur contexte. Pour plus d'informations, voir «Cas où les caractères génériques basés sur des rubriques ne sont pas génériques», à la page 77.

**Remarque :** Le début et la fin d'une chaîne de rubrique sont traités de manière spéciale. Si vous utilisez '\$' pour indiquer la fin de la chaîne, '\$#/...' est un caractère générique à plusieurs niveaux et '\$/#/...' est un noeud vide à la racine, suivi par un caractère générique multi-niveau.





## Cas où les caractères génériques basés sur des rubriques ne sont pas génériques

Les caractères génériques '+' et '#' n'ont pas de signification spéciale lorsqu'ils sont mélangés à d'autres caractères (y compris eux-mêmes) dans un niveau de rubrique.

Cela signifie que les rubriques qui contiennent '+' ou '#' avec d'autres caractères dans un niveau de rubrique peuvent être publiées.

Etudions par exemple les deux rubriques suivantes :

1. level0/level1+/level4/#
2. level0/level1/#+/level4/level#

Dans le premier exemple, les caractères '+' et '#' sont traités comme des caractères génériques et ne sont donc pas valides dans une chaîne de rubrique cible de la publication, mais ils sont valides dans un abonnement.

Dans le second exemple, les caractères '+' et '#' ne sont pas traités comme des caractères génériques et la chaîne de rubrique peut donc être la cible d'une publication et d'un abonnement.

## Exemples

```
IBM/+/Results
#/Results
IBM/Software/Results
```

### *Schéma de caractères génériques (de type caractère)*

Le schéma de caractères génériques (de type caractère) vous permet de sélectionner des rubriques en fonction de la correspondance de caractères standard.

Vous pouvez sélectionner toutes les rubriques à plusieurs niveaux dans une hiérarchie de rubriques à l'aide de la chaîne '\*' . L'utilisation de '\*' dans le schéma de caractères génériques (de type caractère) est équivalente à l'utilisation de la chaîne de caractères génériques basée sur des rubriques '#' .

'x\*/y' est équivalent à 'x/#/y' dans le schéma basé sur des rubriques et sélectionne toutes les rubriques de la hiérarchie de rubriques entre les niveaux 'x' et 'y' , où 'x' et 'y' sont des noms de rubrique qui ne figurent pas dans la série de niveaux renvoyée par le caractère générique.

'/+/ ' dans le schéma basé sur les rubriques n'a pas d'équivalent exact dans le schéma basé sur les caractères. 'IBM\*/Results' sélectionnera également 'IBM/Patents/Software/Results' . Vous pouvez construire des requêtes avec les deux schémas qui renvoient des correspondances identiques uniquement si l'ensemble de noms de rubrique à chaque niveau de la hiérarchie est unique.

En général, '\*' et '?' utilisés dans le schéma basé sur des caractères n'ont pas d'équivalents dans le schéma basé sur des rubriques. Le schéma basé sur des rubriques n'exécute pas de correspondance partielle à l'aide de caractères génériques. L'abonnement générique basé sur des caractères 'IBM/\*ware/Results' ne possède pas d'équivalent basé sur des rubriques.

**Remarque :** Les correspondances utilisant les abonnements génériques (de type caractère) sont plus lentes que les correspondances utilisant les abonnements basés sur des rubriques.

**Chaîne de caractères génériques (de type caractère)**

**Littéral de la version 6**

► N'importe quel caractère Unicode sauf \*,? et% ◄

Remarques :

- <sup>1</sup> Indique de ne pas appliquer le caractère suivant, qui sera traité comme un caractère littéral. '%' doit être suivi de '\*', '?' ou '%'. Voir «Exemples de chaînes de rubrique», à la page 74.
- <sup>2</sup> Signifie "Correspondance avec zéro ou plusieurs caractères" dans un abonnement.
- <sup>3</sup> Signifie "Correspondance exacte avec un seul caractère" dans un abonnement.

## Exemples

```
IBM*/Results
IBM/*ware/Results
```

## Combinaison de chaînes de rubrique

Lors de la création d'abonnement ou de l'ouverture de rubriques pour y publier des messages, la chaîne de rubrique peut être constituée en combinant deux chaînes de sous-rubrique distinctes, ou "sous-rubriques". Une sous-rubrique est fournie par l'application ou la commande d'administration comme chaîne de rubrique, et l'autre est la chaîne de rubrique associée à l'objet de rubrique. Vous pouvez utiliser la sous-rubrique comme chaîne de rubrique seule ou les combiner pour former un nouveau nom de rubrique.

Par exemple, vous pouvez définir un abonnement en utilisant la commande MQSC **DEFINE SUB**. La commande peut prendre **TOPICSTR** (chaîne de rubrique) ou **TOPICOBJ** (objet de rubrique) comme

attribut ou les deux. Si seul **TOPICOBJ** est fourni, la chaîne de rubrique associée à l'objet de rubrique est utilisée comme chaîne de rubrique. Si seul **TOPICSTR** est fourni, il est utilisé comme chaîne de rubrique. Si les deux sont fournis, ils sont concaténés pour former une chaîne de rubrique unique de format **TOPICOBJ / TOPICSTR**, où la chaîne de rubrique configurée **TOPICOBJ** est toujours placée en premier et les deux parties de la chaîne sont toujours séparées par le caractère "/".

De même, dans programme MQI, le nom de rubrique complet est créé par MQOPEN. Il est composé de deux zones utilisées dans des appels MQI de publication/abonnement, dans l'ordre indiqué :

1. L'attribut **TOPICSTR** de l'objet de rubrique, nommé dans la zone **ObjectName**.
2. Le paramètre **ObjectString** définissant la sous-rubrique fournie par l'application.

La chaîne de rubrique résultante est renvoyée dans le paramètre **ResObjectString**.

Ces zones sont considérées comme étant présentes si le premier caractère de chaque zone n'est pas vide ou un caractère nul et si la longueur de la zone est supérieure à zéro. Si seule une zone est présente, elle est utilisée telle quelle comme nom de rubrique. Si aucune zone n'est renseignée, l'appel échoue avec le code anomalie MQRC\_UNKNOWN\_OBJECT\_NAME ou MQRC\_TOPIC\_STRING\_ERROR si le nom de rubrique complet n'est pas valide.

Si les deux zones sont présentes, un caractère "/" est inséré entre les deux éléments du nom de rubrique combiné résultant.

Le [Tableau 2](#), à la page 79 illustre des exemples de concaténation de chaînes de rubrique :

TOPICSTR de l'objet de rubrique	Chaîne de rubrique fournie par l'application ou la commande DEFINE SUB	Nom complet de la rubrique	Commentaire
Football/Scores	' '	Football/Scores	La chaîne TOPICSTR de l'objet de rubrique est utilisée seule.
' '	Football/Scores	Football/Scores	La chaîne ObjectString/ TOPICSTR est utilisée seule.
Football	Scores	Football/Scores	Le caractère "/" est ajouté au point de concaténation.
Football	/Scores	Football//Scores	Un 'noeud vide' est généré entre les deux chaînes. Différent de "Football/Scores".
/Football	Scores	/Football/Scores	La rubrique commence par un 'noeud vide'. Différent de "Football/Scores".

Le caractère "/" est considéré comme un caractère spécial fournissant la structure au nom complet de la rubrique dans in«Arborescence de rubriques», à la page 80. Le caractère "/" ne doit pas être utilisé dans tous les autres cas, car la structure de l'arborescence des rubriques serait affectée. La rubrique "/Football1" est différente de la rubrique "Football1".

**Remarque :** Si vous utilisez un objet de rubrique lors de la création d'un abonnement, la valeur de la chaîne de rubrique de l'objet de rubrique est fixe dans l'abonnement à un moment donné. Toute

modification consécutive de l'objet de rubrique n'affecte pas la chaîne de rubrique où l'abonnement est défini.

## Caractères génériques dans les chaînes de rubrique

Les caractères génériques suivants sont des caractères spéciaux :

- signe plus (+)
- signe dièse (#)
- astérisque (\*)
- point d'interrogation (?)

Les caractères génériques ont une signification spéciale uniquement lorsqu'ils sont utilisés par un abonnement. Ces caractères ne sont pas valides lorsqu'ils sont utilisés autre part, mais vous devez savoir comment ils sont utilisés, et il est peut-être préférable de ne pas les utiliser dans les chaînes de rubrique lors de la publication ou de la définition des objets de rubrique.

Si vous publiez une chaîne de rubrique avec # ou + combiné avec d'autres caractères (y compris eux-mêmes) dans un niveau de rubrique, la chaîne de rubrique peut faire l'objet d'un abonnement avec l'un ou l'autre des schémas de caractère générique.

Si vous publiez une chaîne de rubrique avec # ou + comme seul caractère entre deux caractères /, la chaîne de rubrique ne peut pas faire l'objet d'un abonnement explicitement par une application qui utilise MQSO\_WILDCARD\_TOPIC. Il en résulte que l'application reçoit davantage de publications que prévu.

N'utilisez pas de caractère générique dans la chaîne de rubrique d'un objet de rubrique défini. Dans le cas contraire, le caractère est traité comme caractère littéral lorsque l'objet est utilisé par un diffuseur, et comme caractère générique lorsqu'il est utilisé par un abonnement. Cela peut générer une confusion.

### Exemple de fragment de code

Ce fragment de code, extrait de l'exemple de programme [Exemple 2 : publication dans une rubrique variable](#), combine un objet de rubrique et une chaîne de rubrique variable :

```
MQOD td = {MQOD_DEFAULT}; /* Object Descriptor */
td.ObjectType = MQOT_TOPIC; /* Object is a topic */
td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
td.ObjectString.VSPtr = topicString;
td.ObjectString.VSLength = (MQLONG)strlen(topicString);
td.ResObjectString.VSPtr = resTopicStr;
td.ResObjectString.VSBufSize = sizeof(resTopicStr)-1;
MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF_QUIESCING, &Hobj, &CompCode, &Reason);
```

## Arborescence de rubriques

Chaque rubrique que vous définissez correspond à un élément ou à un noeud de l'arborescence de rubriques. Cette dernière peut être vide pour commencer ou contenir des rubriques définies précédemment à l'aide de commandes MQSC ou PCF. Vous pouvez définir une nouvelle rubrique à l'aide des commandes de création de rubrique ou en spécifiant la rubrique pour la première fois dans une publication ou un abonnement.

Vous pouvez utiliser toute chaîne de caractères pour définir la chaîne de rubrique d'une rubrique, mais il est conseillé de choisir une chaîne de rubrique qui s'intègre dans une structure d'arborescence hiérarchique. Une conception soignée des chaînes de rubrique et des arborescences de rubriques peut vous aider dans le cadre des opérations suivantes :

- Abonnement à plusieurs rubriques.
- Etablissement de règles de sécurité.

Vous pouvez construire une arborescence de rubriques sous la forme d'une structure plate et linéaire, mais il est recommandé de générer une arborescence de rubriques dans une structure hiérarchique

comportant une ou plusieurs rubriques racine. Pour plus d'informations sur la planification de la sécurité et les rubriques, voir [Sécurité de publication/abonnement](#).

La [Figure 18](#), à la page 81 illustre un exemple d'arborescence de rubriques avec une rubrique racine.

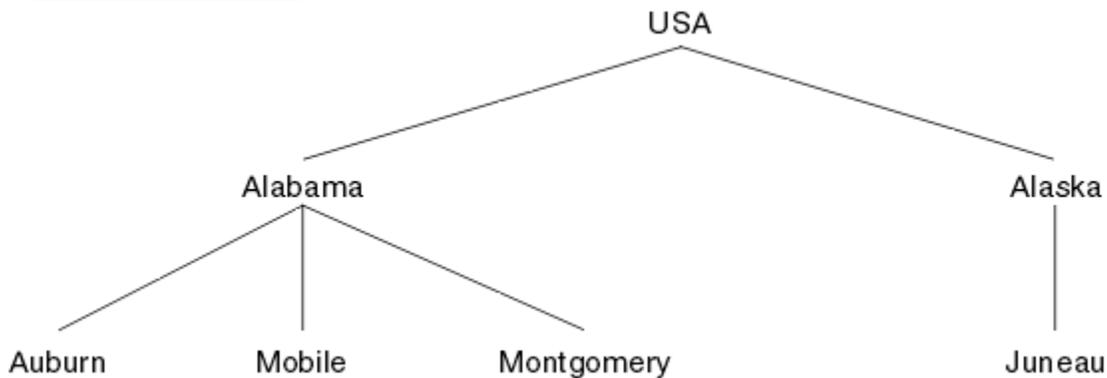


Figure 18. Exemple d'arborescence de rubriques

Chaque chaîne de caractères de la figure représente un noeud de l'arborescence de rubriques. Une chaîne de rubrique complète est créée en agrégeant les noeuds d'un ou plusieurs niveaux de l'arborescence de rubriques. Les niveaux sont séparés par le caractère "/". Le format d'une chaîne de rubrique intégralement spécifiée est le suivant : "racine/niveau2/niveau3".

Les rubriques valides de l'arborescence de rubriques illustrée dans la [Figure 18](#), à la page 81 sont les suivantes :

```
"USA"  
"USA/Alabama"  
"USA/Alaska"  
"USA/Alabama/Auburn"  
"USA/Alabama/Mobile"  
"USA/Alabama/Montgomery"  
"USA/Alaska/Juneau"
```

Lorsque vous concevez des chaînes de rubrique et des arborescences de rubriques, n'oubliez pas que le gestionnaire de files d'attente n'interprète pas la chaîne de rubrique elle-même et qu'il n'essaye pas d'en déterminer la signification. Il utilise simplement la chaîne de rubrique pour envoyer des messages sélectionnés aux abonnés de cette rubrique.

Les principes suivants s'appliquent à la construction et au contenu d'une arborescence de rubriques :

- Le nombre de niveaux d'une arborescence de rubriques est illimité.
- La longueur du nom d'un niveau dans une arborescence de rubriques est illimitée.
- Le nombre de noeuds "racine" est illimité (le nombre d'arborescences de rubriques l'est donc également).

### Tâches associées

[Réduction du nombre de rubriques non souhaitées dans l'arborescence de rubrique](#)

### Objets de rubrique d'administration

A l'aide d'un objet de rubrique d'administration, vous pouvez affecter des attributs spécifiques et autres que ceux par défaut aux rubriques.

[Figure 19](#), à la page 82 illustre comment la rubrique de haut niveau Sport qui est divisée en plusieurs rubriques couvrant divers sports peut être visualisée comme arborescence de rubriques :

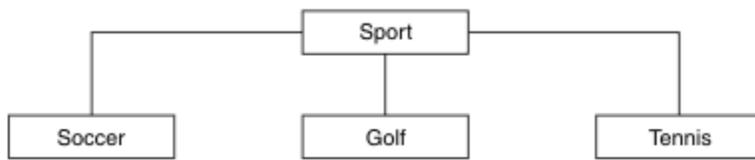


Figure 19. Visualisation d'une arborescence de rubriques

Figure 20, à la page 82 illustre comment l'arborescence de rubriques peut être à nouveau divisée en sous-rubriques traitant de différents types d'informations sur chaque sport :



Figure 20. Arborescence de rubriques étendue

Pour créer l'arborescence de rubriques illustrée, il n'est pas nécessaire de définir des objets de rubrique d'administration. Chaque noeud figurant dans l'arborescence est défini par une chaîne de rubrique créée dans une opération de publication ou d'abonnement. Chaque rubrique de l'arborescence hérite des attributs de son parent. Les attributs sont hérités de l'objet de rubrique parent, car par défaut, tous les attributs sont définis sur `ASPARENT`. Dans cet exemple, toutes les rubriques possèdent les mêmes attributs que la rubrique `Sport`. La rubrique `Sport` ne possède aucun objet de rubrique d'administration et hérite des attributs de `SYSTEM.BASE.TOPIC`.

Notez qu'il est déconseillé d'octroyer des droits d'accès à des utilisateurs non mqm au niveau du noeud racine de l'arborescence de rubriques (`SYSTEM.BASE.TOPIC`), car les droits sont hérités et ne peuvent pas être restreints. De ce fait, en accordant des droits d'accès à ce niveau, vous octroyez des droits d'accès à la totalité de l'arborescence. Vous devez accorder les droits d'accès à un niveau de rubrique inférieur dans la hiérarchie.

Les objets de rubrique d'administration peuvent être utilisés pour définir des attributs spécifiques pour des noeuds particuliers dans l'arborescence de rubriques. Dans l'exemple suivant, l'objet de rubrique d'administration est défini pour attribuer la valeur `NO` à la propriété d'abonnements durables `DURSUB` de la rubrique `Football` :

```

DEFINE TOPIC(FOOTBALL.EUROPEAN)
TOPICSTR('Sport/Soccer')
DURSUB(NO)
DESCR('Administrative topic object to disallow durable subscriptions')
  
```

L'arborescence de rubriques peut être consultée sous forme de :

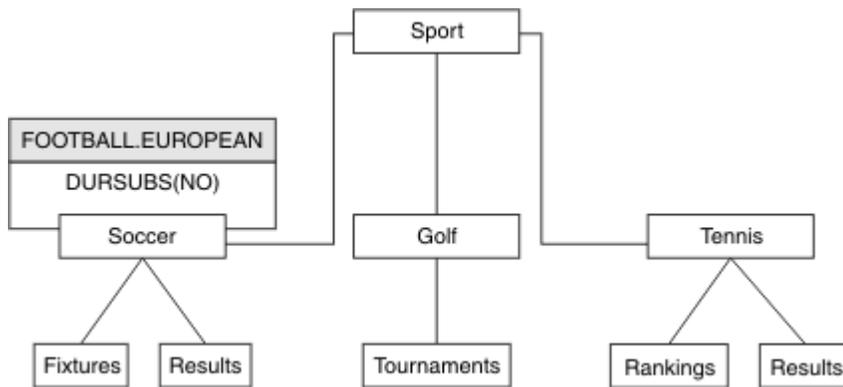


Figure 21. Visualisation d'un objet de rubrique d'administration associé à la rubrique Sport/Football

Toutes les applications qui s'abonnent aux rubriques sous Football dans l'arborescence peuvent toujours utiliser les chaînes de rubrique utilisées avant l'ajout de l'objet de rubrique d'administration. Cependant, une application peut être écrite pour s'abonner à l'aide du nom d'objet FOOTBALL . EUROPEAN au lieu de la chaîne /Sport/Soccer. Par exemple, pour s'abonner à /Sport/Soccer/Results, une application peut spécifier MQSD . ObjectName en tant que FOOTBALL . EUROPEAN et MQSD . ObjectString en tant que Results.

Avec cette fonction, vous pouvez masquer une partie de l'arborescence de rubriques vis-à-vis des développeurs d'applications. Définissez un objet de rubrique d'administration sur un noeud spécifique dans l'arborescence de rubriques, puis les développeurs peuvent définir leurs propres rubriques en tant que rubriques enfant du noeud. Les développeurs doivent uniquement connaître la rubrique parent, et non les autres noeuds de l'arborescence parent.

## Héritage des attributs

Si une arborescence de rubriques comporte plusieurs objets de rubrique d'administration, par défaut chaque objet hérite ses attributs de la rubrique d'administration parent la plus proche. L'exemple précédent a été étendu dans Figure 22, à la page 83 :

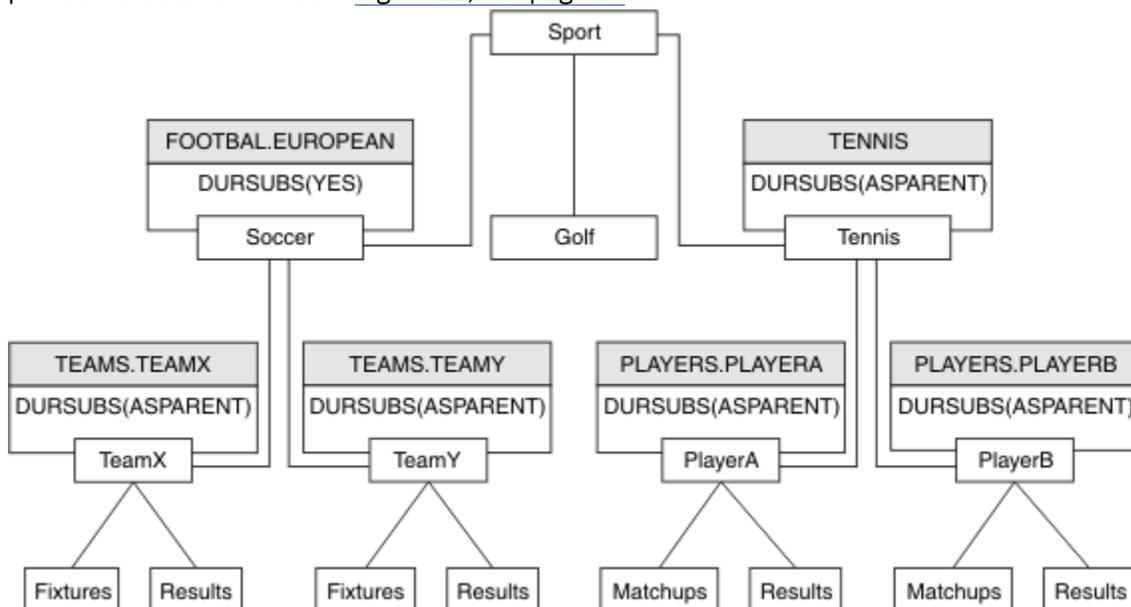


Figure 22. Arborescence de rubriques avec plusieurs objets de rubrique d'administration

Par exemple, utilisez la fonction d'héritage pour attribuer à toutes les rubriques enfant de /Sport/Soccer la propriété où les abonnements sont non durables. Remplacez l'attribut DURSUB de FOOTBALL . EUROPEAN par NO.

Cet attribut peut être défini à l'aide de la commande suivante :

```
ALTER TOPIC(FOOTBALL.EUROPEAN) DURSUB(NO)
```

La propriété DURSUB de tous les objets de rubrique d'administration des rubriques enfant de Sport/Soccer a pour valeur ASPARENT (valeur par défaut). Après avoir modifié la valeur de la propriété DURSUB de FOOTBALL.EUROPEAN en NO, les rubriques enfant de Sport/Soccer héritent de la valeur de propriété DURSUB NO. Toutes les rubriques enfant de Sport/Tennis héritent de la valeur de DURSUB de l'objet SYSTEM.BASE.TOPIC.SYSTEM.BASE.TOPIC est doté de la valeur YES.

La tentative de créer un abonnement durable à la rubrique Sport/Soccer/TeamX/Results va échouer ; cependant, il est possible de créer un abonnement durable à Sport/Tennis/PlayerB/Results.

## Contrôle de l'utilisation de caractères génériques avec la propriété WILDCARD

Utilisez la propriété WILDCARD du paramètre **Topic** de MQSC ou la propriété WildcardOperation équivalente de la commande PCF Topic pour contrôler la distribution des publications sur les applications abonnées qui utilisent les noms de chaîne de rubrique générique. La propriété WILDCARD peut avoir deux valeurs possibles :

### WILDCARD

Comportement des abonnements génériques par rapport à cette rubrique.

#### PASSTHRU

Les abonnements à une rubrique générique moins spécifique que la chaîne de rubrique dans cet objet rubrique reçoivent les publications effectuées dans cette rubrique et les chaînes de rubrique plus spécifiques que cette rubrique.

#### BLOCK

Les abonnements à une rubrique générique moins spécifique que la chaîne de rubrique dans cet objet rubrique ne reçoivent pas les publications effectuées dans cette rubrique ou les chaînes de rubrique plus spécifiques que cette rubrique.

La valeur de cet attribut est utilisée lorsque des abonnements sont définis. Si vous modifiez cet attribut, l'ensemble de rubriques couvert par les abonnements existants n'est pas affecté par la modification. Ce scénario s'applique également si la topologie est modifiée lorsque des objets rubrique sont créés ou supprimés. L'ensemble de rubriques correspondant aux abonnements créés à la suite de la modification de l'attribut WILDCARD est créé en utilisant la topologie modifiée. Si vous voulez forcer la réévaluation de l'ensemble de rubriques correspond pour les abonnements existants, vous devez redémarrer le gestionnaire de files d'attente.

Dans l'exemple, «[Exemple : Créer le cluster de publication/abonnement Sport](#)», à la page 88, vous pouvez suivre les étapes pour créer la structure de l'arborescence de rubriques affichée dans [Figure 23](#), à la page 85.

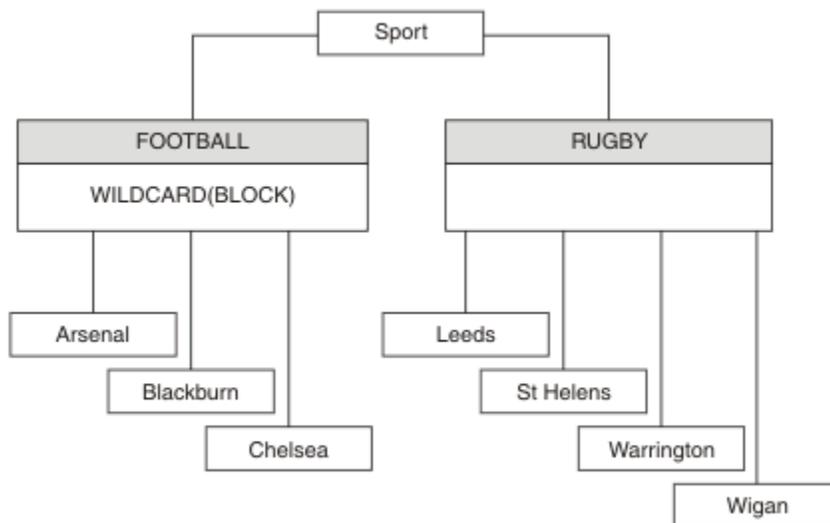


Figure 23. Arborescence de rubriques qui utilise la propriété WILDCARD, BLOCK

Un abonné utilisant la chaîne de rubrique générique # reçoit toutes les publications de la rubrique Sport et de la sous-arborescence Sport/Rugby. L'abonné ne reçoit aucune publication de la sous-arborescence Sport/Football car la valeur de propriété WILDCARD de la rubrique Sport/Football est BLOCK.

PASSTHRU correspond au paramètre par défaut. Vous pouvez définir la valeur PASSTHRU de la propriété WILDCARD sur les nœuds de l'arborescence Sport. Si les nœuds ne disposent pas de la valeur BLOCK de la propriété WILDCARD, la définition de PASSTHRU ne modifie pas le comportement constaté par les abonnés sur les nœuds de l'arborescence Sports.

Dans l'exemple, créez des abonnements pour observer comment le paramètre de caractère générique affecte les publications distribuées ; voir Figure 27, à la page 90. Exécutez la commande de publication dans Figure 30, à la page 91 pour créer quelques publications.

```
pub QMA
```

Figure 24. Publier dans QMA

Les résultats sont affichés dans Tableau 3, à la page 85. Notez comment la définition de la valeur BLOCK de la propriété WILDCARD empêche les abonnements avec caractères génériques de recevoir les publications des rubriques dans la portée du caractère générique.

Tableau 3. Publications reçues sur QMA			
Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Toutes les publications dans la sous-arborescence Football sont bloquées par WILDCARD (BLOCK) dans Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) sur Sports/Football empêche l'abonnement générique dans Arsenal

Tableau 3. Publications reçues sur QMA (suite)

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	La valeur de propriété par défaut WILDCARD dans Sports/Rugby n'empêche pas l'abonnement générique dans Leeds.

**Remarque :**

Supposons qu'un abonnement est doté d'un caractère générique qui fait correspondre un objet de rubrique à la valeur BLOCK de la propriété WILDCARD. Si l'abonnement dispose également d'une chaîne de rubrique située à droite du caractère générique correspondant, il ne reçoit jamais de publication. Les publications qui ne sont pas bloquées sont celles dans les rubriques parent du caractère générique bloqué. Les publications dans les rubriques enfant de la rubrique dotée de la valeur de propriété BLOCK sont bloquées par le caractère générique. Par conséquent, les chaînes de rubrique d'abonnement incluant une rubrique située à droite du caractère générique ne reçoivent jamais de publications à des fins de correspondance.

La définition de la valeur de propriété WILDCARD sur BLOCK ne signifie pas que vous ne pouvez pas créer d'abonnement à l'aide d'une chaîne de rubrique qui comprend des caractères génériques. Un abonnement de ce type est normal. L'abonnement comprend une rubrique explicite qui correspond à la rubrique avec un objet de rubrique dont la propriété WILDCARD a pour valeur BLOCK. Il utilise les caractères génériques pour les rubriques parent ou enfant de la rubrique dont la propriété WILDCARD est définie sur BLOCK. Dans l'exemple figurant dans Figure 23, à la page 85, un abonnement tel que Sports/Football/# peut recevoir des publications.

**Caractères génériques et rubriques de cluster**

Les définitions de rubrique de cluster sont propagées à chaque gestionnaire de files d'attente d'un cluster. L'abonnement à une rubrique de cluster sur un gestionnaire de files d'attente d'un cluster entraîne la création d'abonnements proxy par le gestionnaire. Un abonnement proxy est créé sur chaque gestionnaire de files d'attente du cluster. Les abonnements utilisant les chaînes de rubrique contenant des caractères génériques et associés aux rubriques de cluster peuvent générer un comportement imprévisible. Le comportement est expliqué dans l'exemple qui suit.

Dans la configuration de cluster, par exemple, «Exemple : Créer le cluster de publication/abonnement Sport», à la page 88, QMB dispose du même ensemble d'abonnements que QMA, mais QMB n'a reçu aucune publication après que le diffuseur a publié dans QMA. Voir Figure 24, à la page 85. Même si les rubriques Sports/Football et Sports/Rugby sont des rubriques de cluster, les abonnements définis dans fullsubs.tst ne font pas référence à une rubrique de cluster. Aucun abonnement proxy n'est propagé depuis QMB dans QMA. Sans abonnement proxy, aucune publication sur QMA n'est réacheminée vers QMB.

Certains abonnements, tels Sports/#/Leeds, peuvent sembler référencer une rubrique de cluster, Sports/Rugby dans ce cas. L'abonnement Sports/#/Leeds est résolu en l'objet de rubrique SYSTEM.BASE.TOPIC.

La règle relative à la résolution de l'objet de rubrique référencé par un abonnement tel que Sports/#/Leeds est la suivante. Tronquez la chaîne de rubrique au premier caractère générique. Naviguez vers la gauche dans la chaîne de rubrique et recherchez la première rubrique comprenant un objet de rubrique d'administration associé. L'objet de rubrique peut spécifier un nom de cluster ou définir un objet de rubrique local. Dans l'exemple, Sports/#/Leeds, une fois tronquée, la chaîne de rubrique correspond à Sports, qui ne comprend pas d'objet de rubrique, et par conséquent Sports/#/Leeds hérite de SYSTEM.BASE.TOPIC, qui est un objet de rubrique local.

Pour visualiser comment l'abonnement aux rubriques en cluster peut modifier le fonctionnement de la propagation des caractères génériques, exécutez le script de commande upsubs.bat. Le script efface les files d'attente d'abonnement et ajoute les abonnements de rubrique de cluster dans fullsubs.tst. Exécutez puba.bat de nouveau pour créer un lot de publications ; voir Figure 24, à la page 85.

Tableau 4, à la page 87 affiche le résultat de l'ajout de deux nouveaux abonnements au même gestionnaire de files d'attente sur lequel les publications ont été publiées. Le résultat est comme prévu, les nouvelles publications reçoivent chacune une publication et le nombre de publications reçues par les autres abonnements est inchangé. Les résultats imprévus se produisent sur l'autre gestionnaire de files d'attente de cluster ; voir Tableau 5, à la page 87.

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Toutes les publications dans la sous-arborescence Football sont bloquées par WILDCARD (BLOCK) dans Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) sur Sports/Football empêche l'abonnement générique dans Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	La valeur de propriété par défaut WILDCARD dans Sports/Rugby n'empêche pas l'abonnement générique dans Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal reçoit une publication car l'abonnement ne comporte pas de caractère générique.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds reçoit une publication dans tous les cas.

Tableau 5, à la page 87 affiche le résultat de l'ajout de deux nouveaux abonnements sur QMB et la publication dans QMA. Pour rappel, QMB n'a reçu aucune publication sans ces deux nouveaux abonnements. Comme prévu, les deux nouveaux abonnements reçoivent des publications car Sports/Football et Sports/Rugby sont tous deux des rubriques de cluster. QMB a transmis des abonnements proxy pour Sports/Football/Arsenal et Sports/Rugby/Leeds à QMA, qui a ensuite envoyé les publications à QMB.

Le résultat imprévu est que les deux abonnements Sports/# et Sports/#/Leeds qui ne recevaient pas de publications en reçoivent maintenant. Cette situation est due au fait que les publications Sports/Football/Arsenal et Sports/Rugby/Leeds transférées vers QMB pour les autres abonnements sont maintenant disponibles pour tout abonnement associé à QMB. Par conséquent, les abonnements aux rubriques locales Sports/# et Sports/#/Leeds reçoivent la publication Sports/Rugby/Leeds. Sports/#/Arsenal ne reçoit toujours pas de publication, car la propriété WILDCARD de Sports/Football est définie sur BLOCK.

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SPORTS	Sports/#	Sports/Rugby/ Leeds	Toutes les publications dans la sous-arborescence Football sont bloquées par WILDCARD (BLOCK) dans Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) sur Sports/Football empêche l'abonnement générique dans Arsenal

Tableau 5. Publications reçues sur QMB (suite)

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	La valeur de propriété par défaut WILDCARD dans Sports/Rugby n'empêche pas l'abonnement générique dans Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal reçoit une publication car l'abonnement ne comporte pas de caractère générique.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds reçoit une publication dans tous les cas.

Dans la plupart des applications, il n'est pas recommandé qu'un abonnement influence le comportement d'un autre abonnement. Un des avantages principaux de la propriété WILDCARD dotée de la valeur BLOCK est qu'elle permet aux abonnements de la même chaîne de rubrique contenant des caractères génériques d'avoir le même comportement. Les résultats de l'abonnement sont les mêmes, que l'abonnement soit sur le même gestionnaire de files d'attente que le diffuseur de publications ou sur un gestionnaire différent.

### Caractères génériques et flux

Dans le cadre d'une nouvelle application écrite sur l'API de publication/abonnement, l'effet est qu'un abonnement à \* ne reçoit aucune publication. Pour recevoir toutes les publications Sports, vous devez vous abonner à Sports/\* ou Sports/# ; il en est de même pour les publications Business.

Le comportement d'une application de publication/abonnement en file d'attente existante ne change pas lorsque le courtier de publication/abonnement est migré vers IBM WebSphere MQ 7 et versions ultérieures. La propriété **StreamName** dans les commandes **Publish**, **Register Publisher** ou **Subscriber** est mappée sur le nom de la rubrique vers laquelle le flux a été migré.

### Caractères génériques et points d'abonnement

Dans le cas d'une nouvelle application écrite sur l'API de publication/abonnement, l'effet de la migration est qu'un abonnement à \* ne reçoit aucune publication. Pour recevoir toutes les publications Sports, vous devez vous abonner à Sports/\* ou Sports/# ; il en est de même pour les publications Business.

Le comportement d'une application de publication/abonnement en file d'attente existante ne change pas lorsque le courtier de publication/abonnement est migré vers IBM WebSphere MQ 7 et versions ultérieures. La propriété **SubPoint** dans les commandes **Publish**, **Register Publisher** ou **Subscriber** est mappée sur le nom de la rubrique vers laquelle l'abonnement a été migré.

### Exemple : Créer le cluster de publication/abonnement Sport

Les étapes suivantes créent un cluster, CL1, avec quatre gestionnaires de files d'attente : deux référentiels complets, CL1A et CL1B, et deux référentiels partiels, QMA et QMB. Les référentiels complets sont utilisés pour contenir uniquement les définitions de cluster. QMA est désigné comme hôte de rubrique de cluster. Les abonnements durables sont définis sur QMA et QMB.

**Remarque :** L'exemple est codé pour Windows. Vous devez coder à nouveau [create qmgrs.bat](#) et [create pub.bat](#) pour configurer et tester l'exemple sur les autres plateformes.

1. Créez les fichiers script.
  - a. [Créez topics.tst](#)
  - b. [Créez wildsubs.tst](#)
  - c. [Créez fullsubs.tst](#)

d. Créez `qmgrs.bat`

e. Créez `pub.bat`

2. Exécutez `qmgrs.bat` pour créer la configuration.

```
qmgrs
```

Créez les rubriques dans Figure 23, à la page 85. Le script de la figure 5 crée les rubriques de cluster Sports/Football et Sports/Rugby.

**Remarque :** L'option REPLACE ne remplace pas les propriétés TOPICSTR d'une rubrique. TOPICSTR est une propriété dont la valeur varie dans l'exemple afin de tester les différentes arborescences de rubriques. Pour modifier les rubriques, supprimez d'abord la rubrique.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Figure 25. Supprimer et créer des rubriques : `topics.tst`

**Remarque :** Supprimez les rubriques car REPLACE ne remplace pas les chaînes de rubrique.

Créez des abonnements avec des caractères génériques. Les caractères génériques faisant correspondre les rubriques aux objets de rubrique dans Figure 23, à la page 85. Créez une file d'attente pour chaque abonnement. Les files d'attente sont effacées et les abonnements supprimés lorsque le script est exécuté ou réexécuté.

**Remarque :** L'option REPLACE ne remplace pas les propriétés TOPICOBJ ou TOPICSTR d'un abonnement. TOPICOBJ ou TOPICSTR sont les propriétés dont la valeur varie dans l'exemple afin de tester les différents abonnements. Pour les modifier, supprimez d'abord l'abonnement.

```
DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)
```

Figure 26. Créer des abonnements génériques : `wildsubs.tst`

Créez des abonnements qui font référence aux objets de rubrique du cluster.

**Remarque :**

Le délimiteur, /, est automatiquement inséré entre la chaîne de rubrique référencée par TOPICOBJ et la chaîne de rubrique définie par TOPICSTR.

La définition `DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal')`  
`DEST(QFARSENAL)` crée le même abonnement. TOPICOBJ permet de facilement référencer la chaîne de rubrique déjà définie. Lorsqu'il est créé, l'abonnement ne se rapporte plus à l'objet de rubrique.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)
```

Figure 27. Supprimer et créer des abonnements : `fullsubs.tst`

Créez un cluster avec deux référentiels. Créez deux référentiels partiels pour la publication et l'abonnement. Réexécutez le script pour tout supprimer, puis recommencez. Le script crée également la hiérarchie de rubriques et les abonnements génériques initiaux.

### Remarque :

Sur les autres plateformes, écrivez un script similaire ou tapez toutes les commandes. L'utilisation d'un script permet de tout supprimer rapidement et de recommencer avec une configuration identique.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Figure 28. Créer des gestionnaires de files d'attente : `qmgrs.bat`

Mettez à jour la configuration en ajoutant des abonnements aux rubriques de cluster.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Figure 29. Mettre à jour les abonnements : `upsubs.bat`

Exécutez `pub.bat`, avec un gestionnaire de files d'attente comme paramètre pour publier les messages contenant la chaîne de rubrique de publication. `pub.bat` utilise l'exemple de programme **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Figure 30. Publier : `pub.bat`

## Flux et rubriques

La publication/abonnement en file d'attente propose le concept de flux de publications qui n'existe pas dans le modèle de publication/abonnement intégré. Dans la publication/abonnement en file d'attente, les flux permettent de séparer les flux d'informations des différentes rubriques. A partir d'IBM WebSphere MQ 7.0, un flux est implémenté comme une rubrique de niveau supérieur pouvant être mappée de manière administrative vers un autre identificateur de rubrique.

Le flux par défaut `SYSTEM.BROKER.DEFAULT.STREAM` est configuré automatiquement pour tous les courtiers et les gestionnaires de files d'attente d'un réseau, et aucune configuration supplémentaire n'est requise pour l'utiliser. Imaginez que le flux par défaut est un espace de sujet par défaut sans nom. Les rubriques publiées dans le flux par défaut sont immédiatement disponibles pour tous les gestionnaires de files d'attente connectés depuis IBM WebSphere MQ 7.0, qui s'exécutent avec la publication/l'abonnement en file d'attente activé. Les flux nommés sont similaires aux espaces de sujet nommés séparés. Le flux nommé doit être défini sur chaque courtier sur lequel il est utilisé.

Si les diffuseurs de publications et les abonnés se trouvent sur des gestionnaires de files d'attente différents, une fois qu'ils sont connectés dans la même hiérarchie de courtiers, aucune configuration supplémentaire n'est requise pour que les publications et les abonnements circulent entre eux. La même interopérabilité fonctionne dans l'autre sens également.

## Flux nommés

Un concepteur de solutions qui utilise le modèle de programmation de publication/abonnement en file d'attente peut décider de placer toutes les publications relatives au sport dans un flux nommé `Sport`. Dans IBM WebSphere MQ 6.0, un flux est souvent répliqué automatiquement sur d'autres courtiers qui utilisent la file d'attente modèle, `SYSTEM.BROKER.MODEL.STREAM`. Cependant, pour que le flux soit accessible par un gestionnaire de files d'attente s'exécutant sur la IBM WebSphere MQ 7.0 ou une édition ultérieure, avec la publication/l'abonnement en file d'attente activé, il doit être ajouté manuellement.

Les applications de publication/abonnement en file d'attente qui s'abonnent à `Soccer/Results` dans le flux `Sport` fonctionnent sans modification. Les applications de publication/abonnement intégrées qui s'abonnent à la rubrique `Sport` à l'aide de `MQSUB` et fournissant la chaîne de rubrique `Soccer/Results` reçoivent également les mêmes publications.

La tâche d'ajout d'un flux est décrite dans la rubrique [Ajout d'un flux](#). Il se peut que vous deviez ajouter des flux manuellement pour les deux raisons suivantes.

1. Vous poursuivez le développement de vos applications de publication/abonnement en file d'attente qui s'exécutent sur la version la plus récente du gestionnaire de files d'attente, plutôt que de migrer les applications vers l'interface MQI de publication/abonnement intégrée.
2. Le mappage par défaut des flux sur des rubriques entraîne une "collision" dans l'espace de sujet, et les publications dans un flux possèdent la même chaîne de rubrique que les publications d'un autre flux.

## Droits

Par défaut, la racine de l'arborescence de rubriques contient plusieurs objets de rubrique : `SYSTEM.BASE.TOPIC`, `SYSTEM.BROKER.DEFAULT.STREAM` et `SYSTEM.BROKER.DEFAULT.SUBPOINT`. Les droits (pour la publication ou l'abonnement par exemple) sont déterminés par les droits dans `SYSTEM.BASE.TOPIC` ; tous les droits dans

SYSTEM.BROKER.DEFAULT.STREAM ou SYSTEM.BROKER.DEFAULT.SUBPOINT sont ignorés. Si SYSTEM.BROKER.DEFAULT.STREAM ou SYSTEM.BROKER.DEFAULT.SUBPOINT est supprimé et recréé avec une chaîne de rubrique non vide, les droits définis sur ces objets sont utilisés comme pour un objet de rubrique normal.

## Mappage entre les flux et les rubriques

Un flux de publication/abonnement est imité dans la IBM WebSphere MQ 7.0 et versions ultérieures en créant une file d'attente et en lui attribuant le même nom que le flux. La file d'attente est parfois appelée la file d'attente de flux car elle s'affiche comme telle pour les applications de publication/abonnement en file d'attente. Elle est identifiée par le moteur de publication/abonnement en l'ajoutant à la liste de noms spéciale appelée SYSTEM.QPUBSUB.QUEUE.NAMELIST. Vous pouvez ajouter le nombre de flux requis en ajoutant des files d'attente spéciales additionnelles à la liste de noms. Enfin, vous devez ajouter des rubriques ayant le même nom que les flux, et les mêmes chaînes de rubrique que le nom de flux pour pouvoir publier et vous abonner aux rubriques.

Dans des circonstances exceptionnelles toutefois, vous pouvez attribuer aux rubriques qui correspondent aux flux les chaînes de rubrique de votre choix lorsque vous définissez les rubriques. L'objectif d'une chaîne de rubrique est d'affecter à la rubrique un nom unique dans l'espace de sujet. En règle générale, le nom de flux remplit cette fonction parfaitement. Il arrive parfois qu'un nom de flux entre en conflit avec un nom de rubrique existant. Pour résoudre le problème, choisissez une autre chaîne de rubrique pour la rubrique associée au flux. Sélectionnez n'importe quelle chaîne de rubrique en vous assurant qu'elle est unique.

La chaîne de rubrique définie dans la définition de rubrique est ajoutée comme préfixe à la chaîne de rubrique fournie par les diffuseurs de publications et les abonnés à l'aide des appels MQI MQOPEN ou MQSUB. Les applications se référant aux rubriques à l'aide des objets de rubrique ne sont pas affectées par le choix d'une chaîne de rubrique en préfixe ; vous pouvez ainsi choisir une chaîne de rubrique qui rend les publications uniques dans l'espace de sujet.

Le remappage des différents flux sur les différentes rubriques dépend des préfixes utilisés pour rendre les chaînes de rubrique uniques, afin de distinguer un ensemble de rubriques d'un autre. Vous devez définir une convention d'attribution de nom de rubrique universelle qui est suivie à la lettre pour que le mappage fonctionne.

Dans la IBM WebSphere MQ 7.0, s'il existe une collision de chaînes de rubrique, vous pouvez utiliser les flux pour séparer les espaces de sujet.

Depuis IBM WebSphere MQ 7.0, le mécanisme d'attribution de préfixe est utilisé pour remapper une chaîne de rubrique vers un autre emplacement dans l'espace de sujet.

**Remarque :** Lorsque vous supprimez un flux, supprimez d'abord tous les abonnements du flux. Cette action est très importante si les abonnements proviennent d'autres courtiers de la hiérarchie.

### Exemple

Dans [Figure 31](#), à la page 93, la rubrique ' Sport ' fait que la chaîne de rubrique ' xyz ' génère des publications provenant du flux ' Sport ' et dont le préfixe est ' xyz ' dans l'espace de sujet du gestionnaire de files d'attente de la version 7. La publication ou l'abonnement dans la version 7 à la rubrique ' Sport ' ajoute le préfixe ' xyz ' à la chaîne de rubrique. Si la publication est propagée à un abonné IBM WebSphere MQ 6, le préfixe ' xyz ' est supprimé de la publication et placé dans le flux ' Sport '. Inversement, lorsque la publication est propagée d'IBM WebSphere MQ 6 vers IBM WebSphere MQ 7, du flux ' Sport ' vers la rubrique ' Sport ', le préfixe ' xyz ' est ajouté à la chaîne de rubrique.

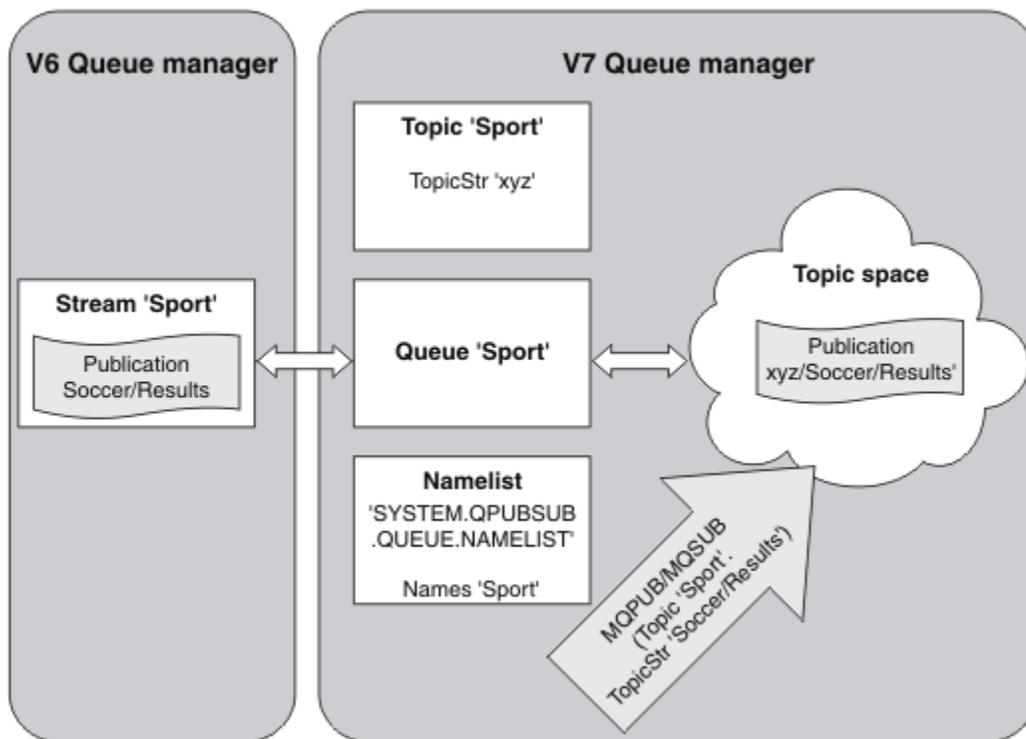


Figure 31. Flux IBM WebSphere MQ 6 qui coexistent avec les rubriques IBM WebSphere MQ 7

### Point d'abonnement et rubriques

Les points d'abonnement nommés sont émulés par les rubriques et les objets de rubrique.

Pour ajouter les points d'abonnement manuellement, voir [Ajout d'un point d'abonnement](#).

### Points d'abonnement dans IBM MQ

IBM MQ mappe les points d'abonnement sur différents espaces de sujet dans l'arborescence de rubriques d'IBM MQ. Les rubriques des messages de commande sans point d'abonnement sont associées à la racine de l'arborescence des rubriques IBM MQ et héritent des propriétés de `SYSTEM.BASE.TOPIC`.

Les messages de commande avec un point d'abonnement sont traités à l'aide de la liste des objets de rubrique dans `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`. Le nom de point d'abonnement dans le message de commande est comparé à la chaîne de rubrique de chaque objet de rubrique dans la liste. S'il existe une correspondance, le nom de point d'abonnement est ajouté en préfixe comme noeud de rubrique à la chaîne de rubrique. La rubrique hérite ses propriétés de l'objet de rubrique associé situé dans `SYSTEM.QPUBSUB.SUBPOINT.NAMELIST`.

Les points d'abonnement sont utilisés pour créer un espace de sujet séparé pour chaque point d'abonnement. L'espace de sujet est associé à la racine d'une rubrique ayant le même nom que le point d'abonnement. Les rubriques dans chaque espace de sujet héritent de leurs propriétés de l'objet de rubrique ayant le même nom que le point d'abonnement.

Toutes les propriétés qui ne sont pas définies dans l'objet de rubrique correspondant sont héritées, de la manière normale, à partir de `SYSTEM.BASE.TOPIC`.

Les applications de publication/abonnement en file d'attente existantes qui utilisent les en-têtes de message `MQRFH2` continuent de s'exécuter en définissant la propriété **SubPoint** dans les messages de commande `Publish` ou `Register subscriber`. Le point d'abonnement est associé à la chaîne de rubrique dans le message de commande et la rubrique obtenue est traitée de la même façon qu'une autre rubrique.

Les applications IBM WebSphere MQ 7.0, ou version ultérieure, ne sont pas affectées par les points d'abonnement. Si une application utilise une rubrique qui hérite de l'un des objets de rubrique correspondants, elle interagit avec une application en file d'attente à l'aide du point d'abonnement correspondant.

### Exemple

Une application de publication/abonnement WebSphere Message Broker existante (également appelée IBM Integration Bus) migrée vers IBM MQ a créé deux objets de rubrique, GBP et USD, avec les chaînes de rubrique correspondantes, 'GBP' et 'USD'.

Les éditeurs existants dans la rubrique NYSE/IBM/SPOT, migrés pour s'exécuter sous IBM MQ, qui utilisent le point d'abonnement USD créent des publications sur la rubrique USD/NYSE/IBM/SPOT. De même, les abonnés existants à NYSE/IBM/SPOT, à l'aide du point d'abonnement USD, créent des abonnements à USD/NYSE/IBM/SPOT.

Abonnez-vous au prix au comptant en dollar dans un programme de publication/abonnement d'IBM WebSphere MQ 7.0, ou version ultérieure, en appelant MQSUB. Créez un abonnement à l'aide de l'objet de rubrique USD et de la chaîne de rubrique 'NYSE/IBM/SPOT', comme illustré dans le fragment de code 'C'.

```
strncpy(sd.ObjectName, "USD", MQ_TOPIC_NAME_LENGTH);
sd.ObjectString.VSPtr = "NYSE/IBM/SPOT";
sd.ObjectString.VSLength = MQVS_NULL_TERMINATED;
MQSUB(Hconn, &sd, &Hobj, &Hsub, &CompCode, &Reason);
```

1. Définissez l'attribut CLUSTER des objets de rubrique USD et GBP sur l'hôte de rubrique de cluster.
2. Supprimez toutes les copies des objets de rubrique USD et GBP sur les autres gestionnaires de files d'attente du cluster.
3. Assurez-vous que USD et GBP sont définis dans SYSTEM.QPUBSUB.SUBPOINT.NAMELIST sur chaque gestionnaire de files d'attente du cluster.

## Exemple de configuration de publication/abonnement à un seul gestionnaire de files d'attente

La Figure 32, à la page 95 illustre une configuration de publication/abonnement de base à un seul gestionnaire de files d'attente. L'exemple montre la configuration d'un service d'information dans lequel des informations sont disponibles à partir de diffuseurs de publications à propos de plusieurs rubriques.

- Diffuseur 1 publie des informations sur les résultats sportifs à l'aide d'une rubrique Sport
- Diffuseur 2 publie des informations sur le cours de la bourse à l'aide d'une rubrique Bourse
- Diffuseur 3 publie des informations sur des critiques de film à l'aide d'une rubrique Films et des programmes de télévision à l'aide d'une rubrique TV

Trois abonnés ont enregistré leur intérêt pour différentes rubriques. Le gestionnaire de files d'attente leur envoie donc les informations qui les intéressent :

- Abonné 1 reçoit les résultats sportifs et les cours de la bourse
- Abonné 2 reçoit les critiques de film
- Abonné 3 reçoit les résultats sportifs

Aucun des abonnés n'a enregistré son intérêt dans les programmes de télévision. Ces informations ne sont donc pas distribuées.

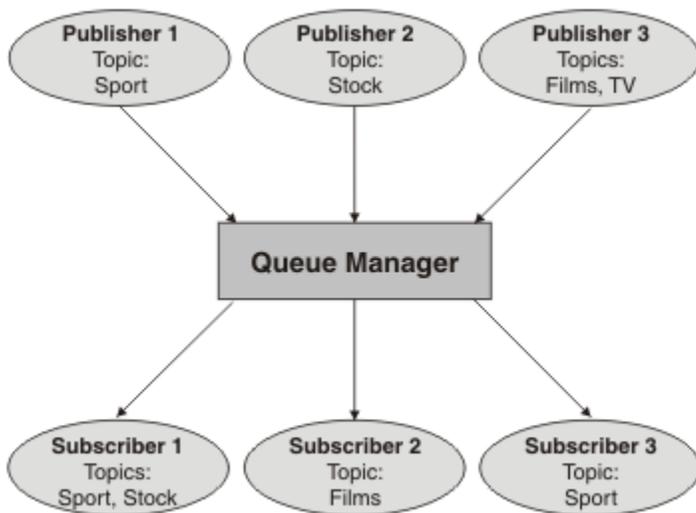


Figure 32. Exemple de publication/abonnement avec un seul gestionnaire de files d'attente

## Réseaux de publication/abonnement répartis

Chaque gestionnaire de files d'attente fait concorder les messages publiés dans une rubrique avec les abonnements créés localement qui se sont abonnés à cette rubrique. Vous pouvez configurer un réseau de gestionnaires de files d'attente pour que les messages publiés par une application connectée à un gestionnaire de files d'attente soient distribués aux abonnements correspondants créés sur d'autres gestionnaires de files d'attente du réseau. Cela nécessite une configuration supplémentaire via des canaux simples entre les gestionnaires de files d'attente.

Une configuration de publication/abonnement répartie est un ensemble de gestionnaires de files d'attente connectés les uns avec les autres. Les gestionnaires de files d'attente peuvent tous se trouver sur le même système physique, ou être répartis sur plusieurs systèmes physiques. Lorsque vous connectez des gestionnaires de files d'attente ensemble, les abonnés peuvent s'abonner à un gestionnaire de files d'attente et recevoir des messages initialement publiés sur un autre gestionnaire de files d'attente. La figure suivante illustre cela en ajoutant un second gestionnaire de files d'attente à la configuration décrite dans «Exemple de configuration de publication/abonnement à un seul gestionnaire de files d'attente», à la page 94.

- Le gestionnaire de files d'attente 2 est utilisé par le diffuseur de publications 4 pour publier des informations relatives aux prévisions météo à l'aide d'une rubrique appelée Météo et des informations sur les conditions de trafic sur les routes à priorité à l'aide d'une rubrique appelée Trafic.
- L'abonné 4 utilise également ce gestionnaire de files d'attente et s'abonne aux informations sur les conditions de trafic à l'aide de la rubrique Trafic.
- L'abonné 3 s'abonne également aux informations relatives à la météo, bien qu'il utilise un gestionnaire de files d'attente différent de celui du diffuseur de publications. Cela est possible car les gestionnaires de files d'attente sont reliés entre eux.

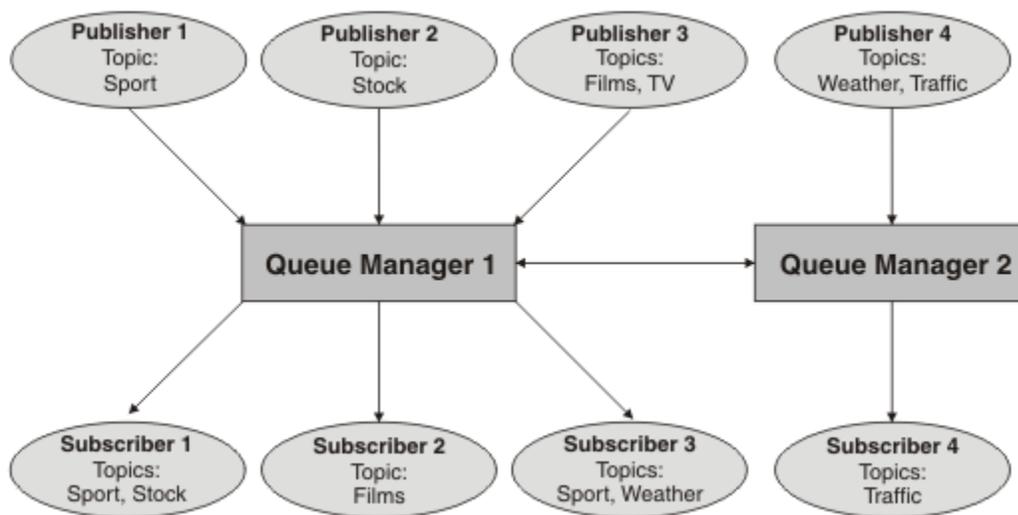


Figure 33. Exemple de publication/abonnement avec deux gestionnaires de files d'attente

Vous pouvez connecter manuellement les gestionnaires de files d'attente dans une hiérarchie parent et enfant, ou créer un cluster de publication/abonnement et laisser IBM MQ définir la plupart des informations de connexion automatiquement. Vous pouvez aussi combiner les deux topologies, par exemple en joignant plusieurs clusters dans une hiérarchie.

### Présentation des clusters de publication/abonnement

Un cluster de publication/abonnement est un cluster standard comprenant un ou plusieurs objets de rubrique. Lorsque vous définissez un objet de rubrique d'administration sur n'importe quel gestionnaire de files d'attente d'un cluster, et que vous le mettez en cluster en indiquant un nom de cluster, les diffuseurs de publications et les abonnés à la rubrique peuvent se connecter à n'importe quel gestionnaire de files d'attente du cluster, et les messages publiés sont routés aux abonnés via les canaux du cluster entre les gestionnaires de files d'attente.

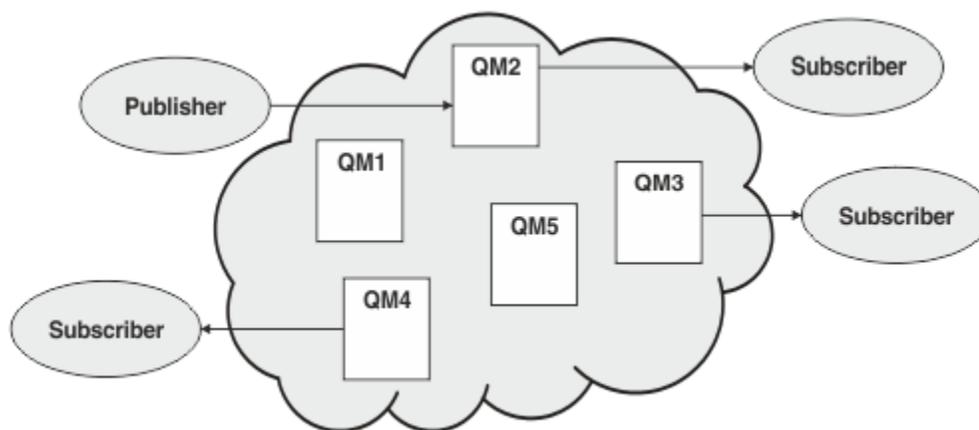


Figure 34. Cluster de publication/abonnement

Vous pouvez configurer le routage des messages de publication/abonnement dans un cluster de deux façons :

- routage direct
- routage via un hôte de rubrique

Lorsque vous configurez une rubrique en cluster à routage direct, les messages publiés sur le gestionnaire de files d'attente sont envoyés directement depuis ce gestionnaire de files d'attente à chaque abonnement sur n'importe quel gestionnaire de files d'attente du cluster. Le chemin le plus direct

pour les publications peut ainsi être obtenu, mais dans ce cas, tous les gestionnaires de files d'attente du cluster ont connaissance de tous les autres gestionnaires de files d'attente, pouvant ainsi établir des canaux de cluster entre eux.

Lorsque vous utilisez le routage via un hôte de rubrique, les messages publiés sur le gestionnaire de files d'attente sont envoyés à partir de ce dernier vers un gestionnaire de files d'attente qui héberge une définition d'objet de rubrique administré. Ce *gestionnaire de files d'attente hôte de rubrique* route le message à tous les abonnements des autres gestionnaires de files d'attente du cluster. Si les diffuseurs de publications ou les abonnés ne se trouvent pas sur les gestionnaires de files d'attente hôte de rubrique, la route des publications est plus longue. Cependant, l'avantage de cette méthode est que seuls les gestionnaires de files d'attente hôte de rubrique ont connaissance de tous les autres gestionnaires de files d'attente et peuvent ainsi établir des canaux de cluster avec eux.

Pour plus d'informations, voir [«Clusters de publication/abonnement»](#), à la page 98.

## Présentation des hiérarchies de publication/abonnement

Une hiérarchie de publication/abonnement est un ensemble de gestionnaires de files d'attente connectés par des canaux dans une structure hiérarchique. Chaque gestionnaire de files d'attente identifie son gestionnaire de files d'attente *parent* comme décrit dans [Connexion d'un gestionnaire de files d'attente à une hiérarchie de publication/abonnement](#).

Les diffuseurs de publications et les abonnés à une rubrique peuvent se connecter à n'importe quel gestionnaire de files d'attente de la hiérarchie, et les messages sont transmis entre eux via la connectivité des gestionnaires de files d'attente hiérarchiques.

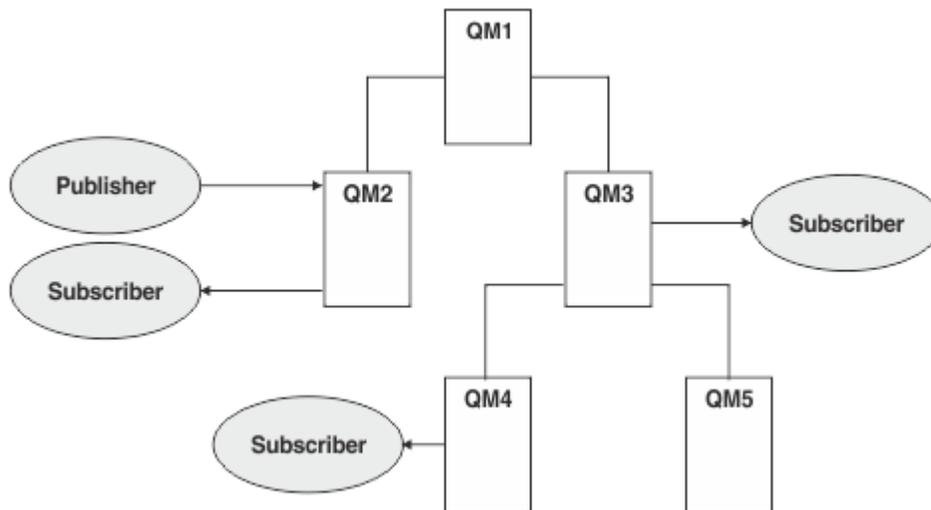


Figure 35. Hiérarchie de publication/abonnement

Dans la figure précédente, les publications distribuées aux abonnés sur QM3 et QM4 étaient routées depuis QM2 vers QM1, puis vers QM3 et finalement vers QM4.

Les hiérarchies permettent de contrôler directement les relations entre chaque gestionnaire de files d'attente de la hiérarchie. Le contrôle du routage des messages depuis les diffuseurs de publications vers les abonnés peut ainsi être plus affiné, surtout lorsque le routage entre les réseaux des gestionnaires de files d'attente s'effectue avec une connectivité restreinte. Accordez une attention particulière à la disponibilité et à la capacité de chaque gestionnaire de files d'attente par lequel un message est routé d'un diffuseur de publications vers les abonnés.

Pour plus d'informations, voir [«Hiérarchies de publication/abonnement»](#), à la page 101.

## Distribution des publications entre les gestionnaires de files d'attente

Outre les choix liés au routage, deux approches permettent de distribuer les publications dans le réseau des gestionnaires de files d'attente :

- Envoyez les publications d'un gestionnaire de files d'attente vers les gestionnaires de files d'attente qui hébergent un abonnement pour cette publication.
- Envoyez chaque publication à tous les gestionnaires de files d'attente, et laissez-les faire correspondre la publication à leurs abonnements.

Dans le premier cas, les messages de publication sont alors uniquement envoyés lorsque cela est nécessaire, mais la connaissance des abonnements doit être partagée entre les gestionnaires de files d'attente. Dans le second cas, la connaissance des abonnements ne doit pas être partagée, mais il se peut que des messages de publication soient envoyés inutilement entre les gestionnaires de files d'attente.

Par défaut, IBM MQ utilise la première méthode et n'envoie les publications qu'aux gestionnaires de files d'attente qui y sont abonnés. La connaissance des abonnements est propagée entre les gestionnaires de files d'attente sous la forme d'*abonnements de proxy*. L'efficacité de la méthode à employer dans une topologie de publication/abonnement dépend de la distribution et de la durée de vie des abonnements, ainsi que de la fréquence des publications. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

### Concepts associés

«Arborescence de rubriques», à la page 80

Chaque rubrique que vous définissez correspond à un élément ou à un noeud de l'arborescence de rubriques. Cette dernière peut être vide pour commencer ou contenir des rubriques définies précédemment à l'aide de commandes MQSC ou PCF. Vous pouvez définir une nouvelle rubrique à l'aide des commandes de création de rubrique ou en spécifiant la rubrique pour la première fois dans une publication ou un abonnement.

### Tâches associées

[Conception des clusters de publication/abonnement](#)

### Référence associée

[Scénarios de hiérarchie de publication/abonnement](#)

## Clusters de publication/abonnement

Un cluster de publication/abonnement est un cluster standard de gestionnaires de files d'attente interconnectés, sur lequel les publications sont automatiquement déplacées des applications de publication vers les abonnements qui existent sur n'importe quel gestionnaire de files d'attente du cluster. Il existe deux options de routage des publications dans un même cluster de publication/abonnement : le *routage direct* et le *routage via un hôte de rubrique*. Le routage que vous choisissez dépend de la taille et des modèles d'activité attendus du cluster.

Un cluster utilisé pour une messagerie de type publication/abonnement n'est pas différent d'un cluster IBM MQ standard. De ce fait, les gestionnaires de files d'attente du cluster de publication/abonnement peuvent exister sur des ordinateurs séparés physiquement et les gestionnaires de chaque paire de gestionnaires de files d'attente sont automatiquement connectés ensemble par des canaux de cluster si nécessaire. Pour plus d'informations, voir [Clusters](#).

Pour configurer un cluster standard de gestionnaires de files d'attente pour la messagerie de publication/abonnement, vous devez définir un ou plusieurs objets gérés sur un gestionnaire de files d'attente du cluster. Pour faire d'une rubrique une rubrique de cluster, vous devez configurer la propriété **CLUSTER** avec le nom du cluster. Toute rubrique alors utilisée par un diffuseur de publications ou par un abonné à ce point ou plus bas dans l'[arborescence de rubriques](#) est partagée par tous les gestionnaires de files d'attente du cluster, et les messages publiés sur une branche de l'arborescence de rubriques sont automatiquement routés vers les abonnements sur d'autres gestionnaires de files d'attente du cluster.

Seule une copie de chaque message est envoyée entre le gestionnaire de files d'attente du diffuseur de publications et chaque autre gestionnaire de files d'attente, quel que soit le nombre d'abonnés au

message sur le gestionnaire de files d'attente cible. A son arrivée sur un gestionnaire de files d'attente avec un ou plusieurs abonnements, le message est dupliqué sur tous les abonnements.

Tout gestionnaire de files d'attente rejoignant le cluster prend automatiquement connaissance des rubriques en cluster, et les diffuseurs de publications ainsi que les abonnés de ce gestionnaire de files d'attente participent automatiquement au cluster.

L'activité de publication/abonnement non mise en cluster peut avoir lieu dans un cluster de publication/abonnement, en utilisant des chaînes de rubrique qui n'entrent dans un objet de rubrique de cluster.

Il existe deux options de routage des publications dans un même cluster de publication/abonnement : le *routage direct* et le *routage via un hôte de rubrique*. Pour choisir le routage du message à utiliser dans le cluster, définissez la propriété **CLROUTE** sur l'objet de rubrique administré à l'une des valeurs suivantes :

- **DIRECT**
- **TOPICHOST**

Par défaut, le routage de rubrique est **DIRECT**. Il s'agissait de l'unique option disponible avant IBM MQ 8.0. Lorsque vous configurez une rubrique de cluster routée directement sur un gestionnaire de files d'attente, tous les gestionnaires de files d'attente du cluster détectent les autres gestionnaires de files d'attente de ce cluster. Lorsqu'il effectue des opérations de publication et d'abonnement, chaque gestionnaire de files d'attente peut se connecter directement à n'importe quel autre gestionnaire de files d'attente du cluster.

A partir de IBM MQ 8.0, vous pouvez à la place configurer le routage de rubrique en tant que **TOPICHOST**. Lorsque vous utilisez le routage via un hôte de rubrique, tous les gestionnaires de files d'attente du cluster détectent les gestionnaires de files d'attente de cluster qui hébergent la définition de rubrique routée (à savoir les gestionnaires de files d'attente sur lesquels vous avez défini l'objet de rubrique). Lorsque vous effectuez des opérations de publication et d'abonnement, les gestionnaires de files d'attente du cluster ne se connectent qu'à ces gestionnaires de files d'attente hôte de rubrique et ne se connectent pas directement les uns aux autres. Les gestionnaires de files d'attente hôte de rubrique sont responsables du routage des publications depuis les gestionnaires de files d'attente sur lesquels les publications sont publiées vers les gestionnaires avec les abonnements correspondants.

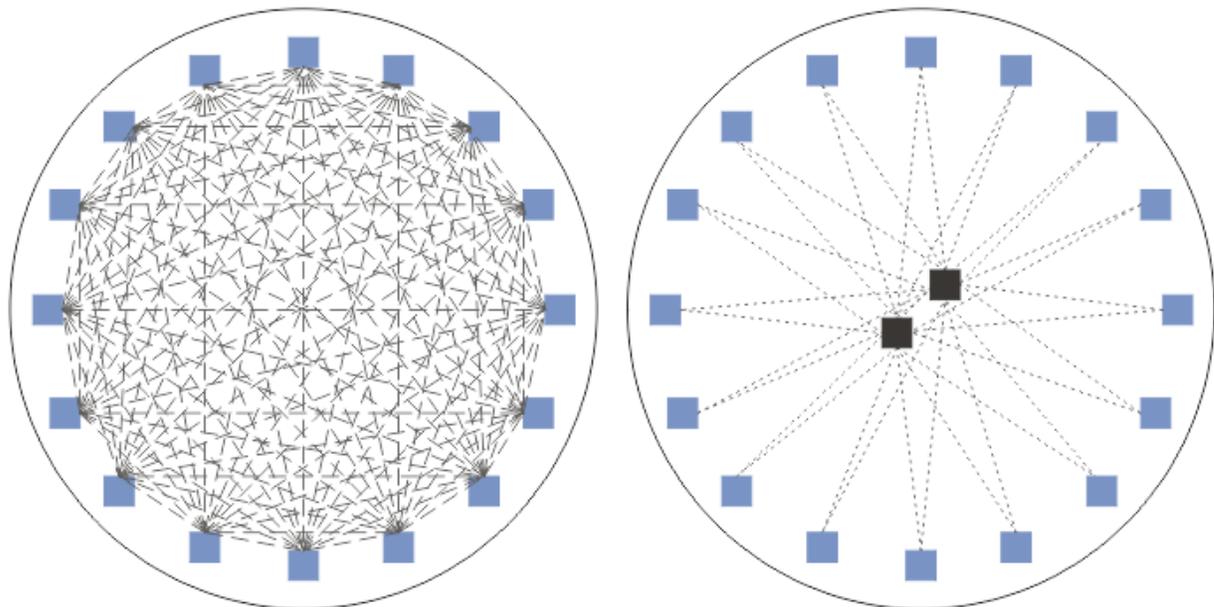


Figure 36. Routage direct et routage via un hôte de rubrique

## Présentation du routage direct

Lorsqu'un objet de rubrique géré est configuré pour le routage direct, il suffit de le définir sur l'un des gestionnaires de files d'attente du cluster pour que tous les gestionnaires de files d'attente le sachent. Le

choix du gestionnaire de files d'attente sur lequel la rubrique est définie n'affecte pas le comportement de la messagerie de publication/abonnement de la rubrique.

Chaque message est transmis directement depuis le gestionnaire de files d'attente du diffuseur de publications vers chaque abonnement sur les autres gestionnaires de files d'attente du cluster, et ne passe par aucun gestionnaire de files d'attente intermédiaire.

Par défaut, les messages sont envoyés uniquement aux autres gestionnaires de files d'attente du cluster qui hébergent un ou plusieurs abonnements.

- Cela implique que chaque gestionnaire de files d'attente informe directement tous les autres gestionnaires de files d'attente du cluster de la présence de toutes les rubriques dotées d'un ou de plusieurs abonnements qui lui sont associés. Tous les gestionnaires de files d'attente du cluster ont ainsi connaissance de toutes les rubriques associées à des abonnements, et de tous les gestionnaires de files d'attente qui hébergent un abonnement établissant un canal vers chaque autre gestionnaire de files d'attente. Cela ne dépend pas du fait que chaque gestionnaire de files d'attente ait un diffuseur de publications ou non.
- La connaissance de chaque rubrique abonnée individuelle sur tous les gestionnaires de files d'attente peut être supprimée par la modification d'un modèle d'envoi de toutes les publications à tous les gestionnaires de files d'attente du cluster, qu'ils soient dotés d'abonnements ou non. Cela réduit le trafic liée à la connaissance des abonnements, mais risque d'augmenter le trafic des publications et le nombre de canaux établis par chaque gestionnaire de files d'attente. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

Les flux de messages de publication/abonnement qui utilisent les rubriques en cluster à routage direct peuvent s'étendre sur plusieurs clusters grâce à l'ajout d'un gestionnaire de files d'attente de chaque cluster dans une hiérarchie de publication/abonnement. Voir [Combinaison d'espaces de sujet de plusieurs clusters](#).

Pour plus d'informations sur le routage direct, voir [Routage direct dans les clusters de publication/abonnement](#).

## Présentation du routage via un hôte de rubrique

Lorsqu'un objet de rubrique administré est configuré pour le routage hôte de rubrique, les publications d'un gestionnaire de files d'attente dans le cluster sont routées via un gestionnaire de files d'attente, où l'objet de rubrique est configuré ("hôte de rubrique"), puis de ce point vers les gestionnaires de files d'attente où se trouvent les abonnements.

- Cela implique que chaque gestionnaire de files d'attente informe tous les hôtes de rubrique de la présence de chaque rubrique ayant un ou plusieurs abonnements qui lui sont associés. N'importe quel gestionnaire de files d'attente hébergeant un abonnement établit un canal vers chaque hôte de rubrique auquel l'abonnement est associé.
- Les gestionnaires de files d'attente qui n'hébergent pas de rubrique n'ont pas connaissance des autres gestionnaires de files d'attente n'hébergeant pas de rubrique dans le cluster à des fins de publication/abonnement et aucun canal n'est établi entre eux dans ce but.
- Si l'application de publication est connectée à un gestionnaire de files d'attente hébergeant la rubrique, les messages publiés sont routés directement aux gestionnaires de files d'attente dans lesquels des abonnements correspondants ont été créés, sans "tronçon" supplémentaire. De même, si les abonnements correspondants sont créés sur le seul gestionnaire de files d'attente hébergeant la rubrique, les messages publiés sur cette rubrique sont routés directement vers ce gestionnaire de files d'attente, sans tronçon supplémentaire.
- Les abonnements figurant sur le même gestionnaire de files d'attente que le diffuseur de publications sont satisfaits sans qu'il soit nécessaire d'effectuer un routage des publications vers les hôtes de l'objet de rubrique.

Tout comme pour les files d'attente de cluster, plusieurs gestionnaires de files d'attente peuvent configurer le même objet de rubrique géré. Cela offre une disponibilité plus élevée du routage des messages et une mise à l'échelle horizontale via l'équilibrage de la charge de travail. Dans le cas d'objets de rubrique routés via un hôte de rubrique, lorsque plusieurs gestionnaires de files d'attente configurent

la même rubrique nommée pour la même branche de l'arborescence de rubriques, chaque hôte de rubrique a connaissance des rubriques abonnées par chaque gestionnaire de files d'attente hébergeant un abonnement.

- Lorsqu'un message est publié, il est envoyé à l'un des gestionnaires de files d'attente hôte de rubrique pour être transmis à l'un des gestionnaires de files d'attente hébergeant l'abonnement. Le choix du gestionnaire de files d'attente hôte de rubrique suit les mêmes règles d'équilibrage de la charge de travail par défaut que pour les files d'attente point-à-point en cluster.
- Si un ou plusieurs gestionnaires de files d'attente hôte de rubrique ne peuvent pas être contacté par un gestionnaire de files d'attente de publication, les messages sont routés vers les gestionnaires de files d'attente hébergeant des rubriques restants disponibles.

Chaque publication vers une rubrique dans une branche routée de l'arborescence de rubriques est transmise à l'un des hôtes de rubrique, même s'il n'existe aucun abonnement à cette rubrique dans le cluster. Par défaut, les messages sont envoyés d'ici uniquement aux autres gestionnaires de files d'attente du cluster qui hébergent un ou plusieurs abonnements.

- Cela implique que chaque gestionnaire de files d'attente hôte de rubrique soit informé de la présence de toutes les chaînes de rubrique abonnées sur chaque gestionnaire de files d'attente du cluster.
- La connaissance de chaque rubrique abonnée individuelle peut être supprimée par la modification d'un modèle d'envoi de toutes les publications à un hôte de rubrique sur tous les gestionnaires de files d'attente du cluster, qu'ils soient ou non dotés d'abonnements. Cela réduit le trafic liée à la connaissance des abonnements, mais risque d'augmenter le trafic des publications et éventuellement le nombre de canaux établis avec chaque gestionnaire de files d'attente hébergeant des rubriques. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

Les flux de messages de publication/abonnement utilisant des rubriques en cluster routées via un hôte de rubrique **ne peuvent pas** s'étendre sur plusieurs clusters de publication/abonnement via l'utilisation d'une hiérarchie de publication/abonnement.

Pour plus d'informations sur le routage via un hôte de rubrique, voir [Routage via un hôte de rubrique dans les clusters de publication/abonnement](#).

## Hiérarchies de publication/abonnement

Pour générer une hiérarchie de publication/abonnement, vous devez lier les gestionnaires de files d'attente entre eux à l'aide de canaux, puis définir une relation parent-enfant entre les paires de gestionnaires de files d'attente. Un message est transmis d'un diffuseur de publications vers les abonnements via les relations directes d'une hiérarchie. Notez que le messages peut avoir à effectuer plusieurs "tronçons" pour arriver à destination.

Seule une copie du message est envoyée à une paire de gestionnaires de files d'attente, quel que soit le nombre d'abonnés du message sur le gestionnaire de files d'attente cible. A son arrivée sur un gestionnaire de files d'attente avec un ou plusieurs abonnements, le message est dupliqué sur tous les abonnements.

Par défaut, les messages sont envoyés uniquement aux autres gestionnaires de files d'attente de la hiérarchie qui se trouvent sur la route d'un abonnement sur un autre gestionnaire de files d'attente :

- Cela implique que chaque gestionnaire de files d'attente informe chaque relation directe de toutes les rubriques dotées d'un ou de plusieurs abonnements, soit sur ce gestionnaire de files d'attente, soit sur l'une de ses autres relations. Ainsi, tous les gestionnaires de files d'attente de la hiérarchie ont connaissance de toutes les rubriques dotées d'un abonnement.
- Ce comportement peut être modifié de telle sorte que les publications soient toujours envoyées à tous les gestionnaires de files d'attente de la hiérarchie, quel que soit le nombre d'abonnements existants. Il n'est alors plus nécessaire de propager les informations sur les abonnements dans la hiérarchie, mais cela peut augmenter le trafic des publications.

Lorsque vous créez un cluster, vous devez faire attention à ne pas créer de boucle entraînant un cycle infini des messages dans le réseau. Une telle boucle ne peut pas être créée dans une hiérarchie.

Chaque gestionnaire de files d'attente doit avoir un nom unique.

Les flux de message de publication/abonnement peuvent s'étendre sur plusieurs clusters de publication/abonnement. Pour ce faire, ajoutez un gestionnaire de files d'attente à partir de chaque cluster dans une hiérarchie de publication/abonnement.

Pour plus d'informations, voir [Routage dans les hiérarchies de publication/abonnement](#).

## **Abonnements de proxy dans un réseau de publication/abonnement**

Un abonnement de proxy est un abonnement souscrit par un gestionnaire de files d'attente pour des rubriques publiées sur un autre gestionnaire de files d'attente. Un abonnement de proxy est transmis entre les gestionnaires de files d'attente pour chaque chaîne de rubrique souscrite par un abonnement. Vous ne créez pas des abonnements de proxy de manière explicite. Le gestionnaire de files d'attente le fait automatiquement en votre nom.

Vous pouvez connecter ensemble des gestionnaire de files d'attente dans une hiérarchie de publication/abonnement ou dans un cluster de publication/abonnement. Les abonnements de proxy transitent entre les gestionnaires de files d'attente connectés. Avec les abonnements de proxy, les publications à une rubrique créées par un diffuseur de publications connecté à un gestionnaire de files d'attente sont reçues par les abonnés à cette rubrique connectés à d'autres gestionnaires de files d'attente. Voir [«Réseaux de publication/abonnement répartis»](#), à la page 95.

Dans des topologies de publication/abonnement avec plusieurs milliers d'abonnements à des chaînes de rubrique individuelles ou dans lesquels l'existence de ces abonnements peut rapidement changer, le temps système lié à la propagation d'abonnements de proxy doit être pris en compte. Outre l'agrégation automatique décrite dans le reste de cette rubrique, vous pouvez effectuer des modifications de configuration manuelles qui limitent davantage le flux des abonnements et des publications entre les gestionnaires de files d'attente connectés et qui réduisent le temps d'attente avant la propagation vers tous les gestionnaires de files d'attente connectés. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

Les abonnements de proxy ne contiennent aucun sélecteur utilisé par les abonnements locaux, et il est possible de simplifier les chaînes de rubrique d'abonnement qui contiennent des caractères génériques. Les publications peuvent alors correspondre à des abonnements de proxy et non aux abonnements réels, ce qui génère un flux de publication supplémentaire entre les gestionnaires de files d'attente. Le gestionnaire de files d'attente qui héberge les abonnements filtre ces différences pour que les publications supplémentaires ne soient pas renvoyées aux abonnements.

### **Agrégation d'abonnements de proxy**

Les abonnements de proxy sont agrégés à l'aide d'un système d'élimination des doublons. Pour une chaîne de rubrique résolue particulière, un abonnement de proxy est envoyé pour le premier abonnement local ou le premier abonnement de proxy reçu. Les abonnements ultérieurs à la même chaîne de rubrique utilisent l'abonnement de proxy existant.

L'abonnement de proxy est annulé après que le dernier abonnement local ou le dernier abonnement de proxy reçu est annulé.

### **Agrégation de publications**

Lorsqu'il existe plusieurs abonnements à la même chaîne de rubrique sur un gestionnaire de files d'attente, une seule copie de chaque publication correspondant à cette chaîne de rubrique est envoyée depuis d'autres gestionnaires de files d'attente dans une topologie de publication/abonnement. A l'arrivée du message, le gestionnaire de files d'attente local fournit une copie du message à chaque abonnement correspondant.

Il est possible que plusieurs abonnements de proxy correspondent à la chaîne de rubrique d'une seule publication lorsque les abonnements de proxy contiennent des caractères génériques. Si un message est publié sur un gestionnaire de files d'attente qui correspond à plusieurs abonnements de proxy créés par un seul gestionnaire de files d'attente connecté, une seule copie de la publication est transmise au gestionnaire de files d'attente éloignées pour satisfaire ces abonnements de proxy multiples.

## Tâches associées

Détection de boucle dans un réseau de publication/abonnement distribué

### **Caractères génériques dans des abonnements de proxy**

Les abonnements peuvent utiliser des caractères génériques dans les chaînes de rubrique pour les faire correspondre à plusieurs chaînes de rubrique des publications.

Il existe deux schémas de caractères génériques pouvant être utilisés par un abonnement : *basé sur les rubriques* et *basé sur les caractères*. Voir «Schémas de caractères génériques», à la page 74.

Dans IBM WebSphere MQ 7.0 et les versions ultérieures, tous les abonnements de proxy pour les abonnements génériques sont convertis pour utiliser les caractères génériques basés sur les rubriques. Si un caractère générique basé sur les caractères est détecté, il est remplacé par un caractère #, placé près du caractère / le plus proche. Par exemple, /aaa/bbb/c\*d est converti en /aaa/bbb/#. Avec la conversion, les gestionnaires de files d'attente éloignées envoient un peu plus de publications que celles auxquelles on s'est abonné explicitement. Les publications en trop sont éliminées par le gestionnaire de files d'attente local lorsque que celui-ci distribue les publications à ses abonnés locaux.

### **Contrôle de l'utilisation de caractères génériques avec la propriété WILDCARD**

Utilisez la propriété WILDCARD du paramètre **Topic** de MQSC ou la propriété WildcardOperation équivalente de la commande PCF Topic pour contrôler la distribution des publications sur les applications abonnées qui utilisent les noms de chaîne de rubrique générique. La propriété WILDCARD peut avoir deux valeurs possibles :

#### **WILDCARD**

Comportement des abonnements génériques par rapport à cette rubrique.

#### **PASSTHRU**

Les abonnements à une rubrique générique moins spécifique que la chaîne de rubrique dans cet objet rubrique reçoivent les publications effectuées dans cette rubrique et les chaînes de rubrique plus spécifiques que cette rubrique.

#### **BLOCK**

Les abonnements à une rubrique générique moins spécifique que la chaîne de rubrique dans cet objet rubrique ne reçoivent pas les publications effectuées dans cette rubrique ou les chaînes de rubrique plus spécifiques que cette rubrique.

La valeur de cet attribut est utilisée lorsque des abonnements sont définis. Si vous modifiez cet attribut, l'ensemble de rubriques couvert par les abonnements existants n'est pas affecté par la modification. Ce scénario s'applique également si la topologie est modifiée lorsque des objets rubrique sont créés ou supprimés. L'ensemble de rubriques correspondant aux abonnements créés à la suite de la modification de l'attribut WILDCARD est créé en utilisant la topologie modifiée. Si vous voulez forcer la réévaluation de l'ensemble de rubriques correspond pour les abonnements existants, vous devez redémarrer le gestionnaire de files d'attente.

Dans l'exemple, «Exemple : Créer le cluster de publication/abonnement Sport», à la page 88, vous pouvez suivre les étapes pour créer la structure de l'arborescence de rubriques affichée dans Figure 23, à la page 85.

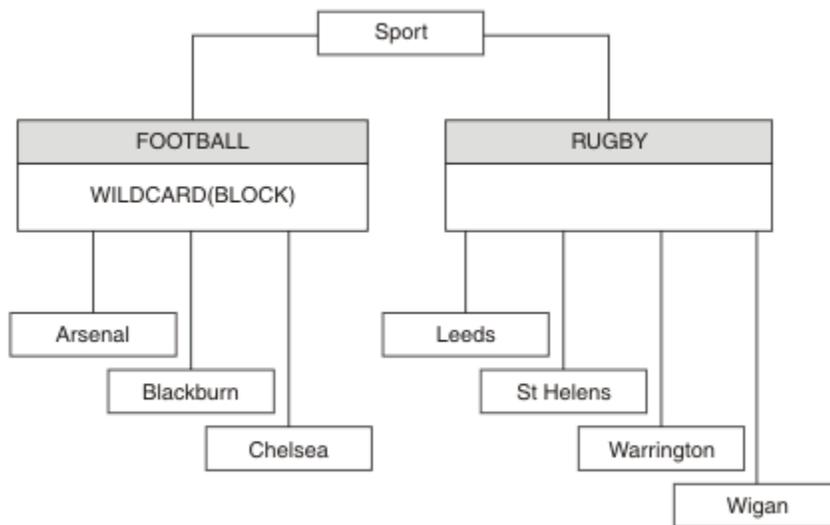


Figure 37. Arborescence de rubriques qui utilise la propriété WILDCARD, BLOCK

Un abonné utilisant la chaîne de rubrique générique # reçoit toutes les publications de la rubrique Sport et de la sous-arborescence Sport/Rugby. L'abonné ne reçoit aucune publication de la sous-arborescence Sport/Football car la valeur de propriété WILDCARD de la rubrique Sport/Football est BLOCK.

PASSTHRU correspond au paramètre par défaut. Vous pouvez définir la valeur PASSTHRU de la propriété WILDCARD sur les nœuds de l'arborescence Sport. Si les nœuds ne disposent pas de la valeur BLOCK de la propriété WILDCARD, la définition de PASSTHRU ne modifie pas le comportement constaté par les abonnés sur les nœuds de l'arborescence Sports.

Dans l'exemple, créez des abonnements pour observer comment le paramètre de caractère générique affecte les publications distribuées ; voir Figure 27, à la page 90. Exécutez la commande de publication dans Figure 30, à la page 91 pour créer quelques publications.

```
pub QMA
```

Figure 38. Publier dans QMA

Les résultats sont affichés dans Tableau 3, à la page 85. Notez comment la définition de la valeur BLOCK de la propriété WILDCARD empêche les abonnements avec caractères génériques de recevoir les publications des rubriques dans la portée du caractère générique.

Tableau 6. Publications reçues sur QMA			
Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Toutes les publications dans la sous-arborescence Football sont bloquées par WILDCARD (BLOCK) dans Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) sur Sports/Football empêche l'abonnement générique dans Arsenal

Tableau 6. Publications reçues sur QMA (suite)

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	La valeur de propriété par défaut WILDCARD dans Sports/Rugby n'empêche pas l'abonnement générique dans Leeds.

**Remarque :**

Supposons qu'un abonnement est doté d'un caractère générique qui fait correspondre un objet de rubrique à la valeur BLOCK de la propriété WILDCARD. Si l'abonnement dispose également d'une chaîne de rubrique située à droite du caractère générique correspondant, il ne reçoit jamais de publication. Les publications qui ne sont pas bloquées sont celles dans les rubriques parent du caractère générique bloqué. Les publications dans les rubriques enfant de la rubrique dotée de la valeur de propriété BLOCK sont bloquées par le caractère générique. Par conséquent, les chaînes de rubrique d'abonnement incluant une rubrique située à droite du caractère générique ne reçoivent jamais de publications à des fins de correspondance.

La définition de la valeur de propriété WILDCARD sur BLOCK ne signifie pas que vous ne pouvez pas créer d'abonnement à l'aide d'une chaîne de rubrique qui comprend des caractères génériques. Un abonnement de ce type est normal. L'abonnement comprend une rubrique explicite qui correspond à la rubrique avec un objet de rubrique dont la propriété WILDCARD a pour valeur BLOCK. Il utilise les caractères génériques pour les rubriques parent ou enfant de la rubrique dont la propriété WILDCARD est définie sur BLOCK. Dans l'exemple figurant dans Figure 23, à la page 85, un abonnement tel que Sports/Football/# peut recevoir des publications.

**Caractères génériques et rubriques de cluster**

Les définitions de rubrique de cluster sont propagées à chaque gestionnaire de files d'attente d'un cluster. L'abonnement à une rubrique de cluster sur un gestionnaire de files d'attente d'un cluster entraîne la création d'abonnements proxy par le gestionnaire. Un abonnement proxy est créé sur chaque gestionnaire de files d'attente du cluster. Les abonnements utilisant les chaînes de rubrique contenant des caractères génériques et associés aux rubriques de cluster peuvent générer un comportement imprévisible. Le comportement est expliqué dans l'exemple qui suit.

Dans la configuration de cluster, par exemple, «Exemple : Créer le cluster de publication/abonnement Sport», à la page 88, QMB dispose du même ensemble d'abonnements que QMA, mais QMB n'a reçu aucune publication après que le diffuseur a publié dans QMA. Voir Figure 24, à la page 85. Même si les rubriques Sports/Football et Sports/Rugby sont des rubriques de cluster, les abonnements définis dans fullsubs.tst ne font pas référence à une rubrique de cluster. Aucun abonnement proxy n'est propagé depuis QMB dans QMA. Sans abonnement proxy, aucune publication sur QMA n'est réacheminée vers QMB.

Certains abonnements, tels Sports/#/Leeds, peuvent sembler référencer une rubrique de cluster, Sports/Rugby dans ce cas. L'abonnement Sports/#/Leeds est résolu en l'objet de rubrique SYSTEM.BASE.TOPIC.

La règle relative à la résolution de l'objet de rubrique référencé par un abonnement tel que Sports/#/Leeds est la suivante. Tronquez la chaîne de rubrique au premier caractère générique. Naviguez vers la gauche dans la chaîne de rubrique et recherchez la première rubrique comprenant un objet de rubrique d'administration associé. L'objet de rubrique peut spécifier un nom de cluster ou définir un objet de rubrique local. Dans l'exemple, Sports/#/Leeds, une fois tronquée, la chaîne de rubrique correspond à Sports, qui ne comprend pas d'objet de rubrique, et par conséquent Sports/#/Leeds hérite de SYSTEM.BASE.TOPIC, qui est un objet de rubrique local.

Pour visualiser comment l'abonnement aux rubriques en cluster peut modifier le fonctionnement de la propagation des caractères génériques, exécutez le script de commande upsubs.bat. Le script efface les files d'attente d'abonnement et ajoute les abonnements de rubrique de cluster dans fullsubs.tst. Exécutez puba.bat de nouveau pour créer un lot de publications ; voir Figure 24, à la page 85.

Tableau 4, à la page 87 affiche le résultat de l'ajout de deux nouveaux abonnements au même gestionnaire de files d'attente sur lequel les publications ont été publiées. Le résultat est comme prévu, les nouvelles publications reçoivent chacune une publication et le nombre de publications reçues par les autres abonnements est inchangé. Les résultats imprévus se produisent sur l'autre gestionnaire de files d'attente de cluster ; voir Tableau 5, à la page 87.

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SPORTS	Sports/#	Sports Sports/Rugby Sports/Rugby/Leeds	Toutes les publications dans la sous-arborescence Football sont bloquées par WILDCARD (BLOCK) dans Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) sur Sports/Football empêche l'abonnement générique dans Arsenal
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	La valeur de propriété par défaut WILDCARD dans Sports/Rugby n'empêche pas l'abonnement générique dans Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal reçoit une publication car l'abonnement ne comporte pas de caractère générique.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds reçoit une publication dans tous les cas.

Tableau 5, à la page 87 affiche le résultat de l'ajout de deux nouveaux abonnements sur QMB et la publication dans QMA. Pour rappel, QMB n'a reçu aucune publication sans ces deux nouveaux abonnements. Comme prévu, les deux nouveaux abonnements reçoivent des publications car Sports/Football et Sports/Rugby sont tous deux des rubriques de cluster. QMB a transmis des abonnements proxy pour Sports/Football/Arsenal et Sports/Rugby/Leeds à QMA, qui a ensuite envoyé les publications à QMB.

Le résultat imprévu est que les deux abonnements Sports/# et Sports/#/Leeds qui ne recevaient pas de publications en reçoivent maintenant. Cette situation est due au fait que les publications Sports/Football/Arsenal et Sports/Rugby/Leeds transférées vers QMB pour les autres abonnements sont maintenant disponibles pour tout abonnement associé à QMB. Par conséquent, les abonnements aux rubriques locales Sports/# et Sports/#/Leeds reçoivent la publication Sports/Rugby/Leeds. Sports/#/Arsenal ne reçoit toujours pas de publication, car la propriété WILDCARD de Sports/Football est définie sur BLOCK.

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SPORTS	Sports/#	Sports/Rugby/ Leeds	Toutes les publications dans la sous-arborescence Football sont bloquées par WILDCARD (BLOCK) dans Sports/Football
SARSENAL	Sports/#/Arsenal	-	WILDCARD (BLOCK) sur Sports/Football empêche l'abonnement générique dans Arsenal

Tableau 8. Publications reçues sur QMB (suite)

Abonnement	Chaîne de rubrique	Publications reçues	Remarques
SLEEDS	Sports/#/Leeds	Sports/Rugby/Leeds	La valeur de propriété par défaut WILDCARD dans Sports/Rugby n'empêche pas l'abonnement générique dans Leeds.
FARSENAL	Sports/Football/Arsenal	Sports/Football/Arsenal	Arsenal reçoit une publication car l'abonnement ne comporte pas de caractère générique.
FLEEDS	Sports/Rugby/Leeds	Sports/Rugby/Leeds	Leeds reçoit une publication dans tous les cas.

Dans la plupart des applications, il n'est pas recommandé qu'un abonnement influence le comportement d'un autre abonnement. Un des avantages principaux de la propriété WILDCARD dotée de la valeur BLOCK est qu'elle permet aux abonnements de la même chaîne de rubrique contenant des caractères génériques d'avoir le même comportement. Les résultats de l'abonnement sont les mêmes, que l'abonnement soit sur le même gestionnaire de files d'attente que le diffuseur de publications ou sur un gestionnaire différent.

### Caractères génériques et flux

Dans le cadre d'une nouvelle application écrite sur l'API de publication/abonnement, l'effet est qu'un abonnement à \* ne reçoit aucune publication. Pour recevoir toutes les publications Sports, vous devez vous abonner à Sports/\* ou Sports/# ; il en est de même pour les publications Business.

Le comportement d'une application de publication/abonnement en file d'attente existante ne change pas lorsque le courtier de publication/abonnement est migré vers IBM WebSphere MQ 7 et versions ultérieures. La propriété **StreamName** dans les commandes **Publish**, **Register Publisher** ou **Subscriber** est mappée sur le nom de la rubrique vers laquelle le flux a été migré.

### Caractères génériques et points d'abonnement

Dans le cas d'une nouvelle application écrite sur l'API de publication/abonnement, l'effet de la migration est qu'un abonnement à \* ne reçoit aucune publication. Pour recevoir toutes les publications Sports, vous devez vous abonner à Sports/\* ou Sports/# ; il en est de même pour les publications Business.

Le comportement d'une application de publication/abonnement en file d'attente existante ne change pas lorsque le courtier de publication/abonnement est migré vers IBM WebSphere MQ 7 et versions ultérieures. La propriété **SubPoint** dans les commandes **Publish**, **Register Publisher** ou **Subscriber** est mappée sur le nom de la rubrique vers laquelle l'abonnement a été migré.

### Exemple : Créer le cluster de publication/abonnement Sport

Les étapes suivantes créent un cluster, CL1, avec quatre gestionnaires de files d'attente : deux référentiels complets, CL1A et CL1B, et deux référentiels partiels, QMA et QMB. Les référentiels complets sont utilisés pour contenir uniquement les définitions de cluster. QMA est désigné comme hôte de rubrique de cluster. Les abonnements durables sont définis sur QMA et QMB.

**Remarque :** L'exemple est codé pour Windows. Vous devez coder à nouveau [create qmgrs.bat](#) et [create pub.bat](#) pour configurer et tester l'exemple sur les autres plateformes.

1. Créez les fichiers script.
  - a. [Créez topics.tst](#)
  - b. [Créez wildsubs.tst](#)
  - c. [Créez fullsubs.tst](#)

d. Créez `qmgrs.bat`

e. Créez `pub.bat`

2. Exécutez `qmgrs.bat` pour créer la configuration.

```
qmgrs
```

Créez les rubriques dans Figure 23, à la page 85. Le script de la figure 5 crée les rubriques de cluster Sports/Football et Sports/Rugby.

**Remarque :** L'option REPLACE ne remplace pas les propriétés TOPICSTR d'une rubrique. TOPICSTR est une propriété dont la valeur varie dans l'exemple afin de tester les différentes arborescences de rubriques. Pour modifier les rubriques, supprimez d'abord la rubrique.

```
DELETE TOPIC ('Sports')
DELETE TOPIC ('Football')
DELETE TOPIC ('Arsenal')
DELETE TOPIC ('Blackburn')
DELETE TOPIC ('Chelsea')
DELETE TOPIC ('Rugby')
DELETE TOPIC ('Leeds')
DELETE TOPIC ('Wigan')
DELETE TOPIC ('Warrington')
DELETE TOPIC ('St. Helens')

DEFINE TOPIC ('Sports') TOPICSTR('Sports')
DEFINE TOPIC ('Football') TOPICSTR('Sports/Football') CLUSTER(CL1) WILDCARD(BLOCK)
DEFINE TOPIC ('Arsenal') TOPICSTR('Sports/Football/Arsenal')
DEFINE TOPIC ('Blackburn') TOPICSTR('Sports/Football/Blackburn')
DEFINE TOPIC ('Chelsea') TOPICSTR('Sports/Football/Chelsea')
DEFINE TOPIC ('Rugby') TOPICSTR('Sports/Rugby') CLUSTER(CL1)
DEFINE TOPIC ('Leeds') TOPICSTR('Sports/Rugby/Leeds')
DEFINE TOPIC ('Wigan') TOPICSTR('Sports/Rugby/Wigan')
DEFINE TOPIC ('Warrington') TOPICSTR('Sports/Rugby/Warrington')
DEFINE TOPIC ('St. Helens') TOPICSTR('Sports/Rugby/St. Helens')
```

Figure 39. Supprimer et créer des rubriques : `topics.tst`

**Remarque :** Supprimez les rubriques car REPLACE ne remplace pas les chaînes de rubrique.

Créez des abonnements avec des caractères génériques. Les caractères génériques faisant correspondre les rubriques aux objets de rubrique dans Figure 23, à la page 85. Créez une file d'attente pour chaque abonnement. Les files d'attente sont effacées et les abonnements supprimés lorsque le script est exécuté ou réexécuté.

**Remarque :** L'option REPLACE ne remplace pas les propriétés TOPICOBJ ou TOPICSTR d'un abonnement. TOPICOBJ ou TOPICSTR sont les propriétés dont la valeur varie dans l'exemple afin de tester les différents abonnements. Pour les modifier, supprimez d'abord l'abonnement.

```
DEFINE QLOCAL(QSPORTS) REPLACE
DEFINE QLOCAL(QSARSENAL) REPLACE
DEFINE QLOCAL(QSLEEDS) REPLACE
CLEAR QLOCAL(QSPORTS)
CLEAR QLOCAL(QSARSENAL)
CLEAR QLOCAL(QSLEEDS)

DELETE SUB (SPORTS)
DELETE SUB (SARSENAL)
DELETE SUB (SLEEDS)
DEFINE SUB (SPORTS) TOPICSTR('Sports/#') DEST(QSPORTS)
DEFINE SUB (SARSENAL) TOPICSTR('Sports+/Arsenal') DEST(QSARSENAL)
DEFINE SUB (SLEEDS) TOPICSTR('Sports+/Leeds') DEST(QSLEEDS)
```

Figure 40. Créer des abonnements génériques : `wildsubs.tst`

Créez des abonnements qui font référence aux objets de rubrique du cluster.

**Remarque :**

Le délimiteur, /, est automatiquement inséré entre la chaîne de rubrique référencée par TOPICOBJ et la chaîne de rubrique définie par TOPICSTR.

La définition DEFINE SUB(FARSENAL) TOPICSTR('Sports/Football/Arsenal') DEST(QFARSENAL) crée le même abonnement. TOPICOBJ permet de facilement référencer la chaîne de rubrique déjà définie. Lorsqu'il est créé, l'abonnement ne se rapporte plus à l'objet de rubrique.

```
DEFINE QLOCAL(QFARSENAL) REPLACE
DEFINE QLOCAL(QRLEEDS) REPLACE
CLEAR QLOCAL(QFARSENAL)
CLEAR QLOCAL(QRLEEDS)

DELETE SUB (FARSENAL)
DELETE SUB (RLEEDS)
DEFINE SUB (FARSENAL) TOPICOBJ('Football') TOPICSTR('Arsenal') DEST(QFARSENAL)
DEFINE SUB (RLEEDS) TOPICOBJ('Rugby') TOPICSTR('Leeds') DEST(QRLEEDS)
```

Figure 41. Supprimer et créer des abonnements : fullsubs.tst

Créez un cluster avec deux référentiels. Créez deux référentiels partiels pour la publication et l'abonnement. Réexécutez le script pour tout supprimer, puis recommencez. Le script crée également la hiérarchie de rubriques et les abonnements génériques initiaux.

#### Remarque :

Sur les autres plateformes, écrivez un script similaire ou tapez toutes les commandes. L'utilisation d'un script permet de tout supprimer rapidement et de recommencer avec une configuration identique.

```
@echo off
set port.CL1B=1421
set port.CL1A=1420
for %%A in (CL1A CL1B QMA QMB) do call :createQM %%A
call :configureQM CL1A CL1B %port.CL1B% full
call :configureQM CL1B CL1A %port.CL1A% full
for %%A in (QMA QMB) do call :configureQM %%A CL1A %port.CL1A% partial
for %%A in (topics.tst wildsubs.tst) do runmqsc QMA < %%A
for %%A in (wildsubs.tst) do runmqsc QMB < %%A
goto:eof

:createQM
echo Configure Queue manager %1
endmqm -p %1
for %%B in (dlt crt str) do %%Bmqm %1
goto:eof

:configureQM
if %1==CL1A set p=1420
if %1==CL1B set p=1421
if %1==QMA set p=1422
if %1==QMB set p=1423
echo configure %1 on port %p% connected to repository %2 on port %3 as %4 repository
echo DEFINE LISTENER(LST%1) TRPTYPE(TCP) PORT(%p%) CONTROL(QMGR) REPLACE | runmqsc %1
echo START LISTENER(LST%1) | runmqsc %1
if full==%4 echo ALTER QMGR REPOS(CL1) DEADQ(SYSTEM.DEAD.LETTER.QUEUE) | runmqsc %1
echo DEFINE CHANNEL(TO.%2) CHLTYPE(CLUSSDR) TRPTYPE(TCP) CONNAME('LOCALHOST(%3)') CLUSTER(CL1)
REPLACE | runmqsc %1
echo DEFINE CHANNEL(TO.%1) CHLTYPE(CLUSRCVR) TRPTYPE(TCP) CONNAME('LOCALHOST(%p%)')
CLUSTER(CL1) REPLACE | runmqsc %1
goto:eof
```

Figure 42. Créer des gestionnaires de files d'attente : qmgrs.bat

Mettez à jour la configuration en ajoutant des abonnements aux rubriques de cluster.

```
@echo off
for %%A in (QMA QMB) do runmqsc %%A < wildsubs.tst
for %%A in (QMA QMB) do runmqsc %%A < upsubs.tst
```

Figure 43. Mettre à jour les abonnements : upsubs.bat

Exécutez `pub.bat`, avec un gestionnaire de files d'attente comme paramètre pour publier les messages contenant la chaîne de rubrique de publication. `pub.bat` utilise l'exemple de programme **amqspub**.

```
@echo off
@rem Provide queue manager name as a parameter
set S=Sports
set S=6 Sports/Football Sports/Football/Arsenal
set S=6 Sports/Rugby Sports/Rugby/Leeds
for %%B in (6) do echo %%B | amqspub %%B %1
```

Figure 44. Publier : `pub.bat`

## Concepts associés

[Abonnements génériques et publications conservées](#)

## Portée de la publication

Lorsque vous configurez un cluster ou une hiérarchie de publication/abonnement, la portée d'une publication détermine également si les gestionnaires de files d'attente transmettent une publication aux gestionnaires de files d'attente éloignées. Utilisez l'attribut de rubrique **PUBSCOPE** pour administrer la portée des publications.

Si une publication n'est pas transmise à des gestionnaires de files d'attente éloignées, seuls les abonnés locaux la reçoivent.

Lorsque vous utilisez un cluster de publication/abonnement, la portée des publications est principalement contrôlée par la définition des objets de rubrique en cluster à certains points de l'arborescence de rubriques. La portée des publications doit être définie pour autoriser le flux des publications vers d'autres gestionnaires de files d'attente du cluster. Vous devez restreindre la portée des publications d'une rubrique en cluster uniquement lorsque vous avez besoin de contrôler avec une granularité fine des rubriques spécifiques sur certains gestionnaires de files d'attente.

Lorsque vous utilisez une hiérarchie de publication/abonnement, la portée des publications est généralement contrôlée par cet attribut, combiné avec l'attribut [Portée des publications](#).

L'attribut **PUBSCOPE** permet de déterminer la portée des publications effectuées sur une rubrique spécifique. Vous pouvez définir cet attribut sur l'une des valeurs suivantes :

### QMGR

La publication est distribuée uniquement aux abonnés locaux. Ces publications sont appelées *publications locales*. Les publications locales sont pas transmises aux gestionnaires de files d'attente éloignées et sont ne donc pas reçues par les abonnés connectés à des gestionnaires de files d'attente éloignées.

### TOUT

La publication est distribuée aux abonnés locaux et aux abonnés connectés à des gestionnaires de files d'attente éloignées dans un cluster ou une hiérarchie de publication/abonnement. Ces publications sont appelées *publications globales*.

### ASPARENT

Utilisez le paramètre **PUBSCOPE** de la rubrique parent dans l'arborescence des rubriques.

Les diffuseurs de publications peuvent également indiquer si une publication est locale ou globale à l'aide de l'option de MQPMO\_SCOPE\_QMGR d'insertion de messages. Si cette option est utilisée, elle remplace le comportement qui a été défini à l'aide de l'attribut de rubrique **PUBSCOPE**.

## Concepts associés

«Objets de rubrique d'administration», à la page 81

A l'aide d'un objet de rubrique d'administration, vous pouvez affecter des attributs spécifiques et autres que ceux par défaut aux rubriques.

## Tâches associées

[Configuration des réseaux de publication/abonnement réparti](#)

## Portée de l'abonnement

La portée d'un abonnement contrôle si un abonnement sur un gestionnaire de files d'attente reçoit des publications qui sont publiées sur un autre gestionnaire de files d'attente dans un cluster ou une hiérarchie de publication/abonnement, ou s'il reçoit uniquement les publications de diffuseurs de publications locaux.

La limitation de la portée d'un abonnement à un gestionnaire de files d'attente arrête la transmission des abonnements de proxy vers d'autres gestionnaires de files d'attente de la topologie de publication/abonnement. Cela réduit le trafic de messagerie de type publication/abonnement entre les gestionnaires de files d'attente.

Lorsque vous utilisez un cluster de publication/abonnement, la portée des abonnements est principalement contrôlée par la définition des objets de rubrique en cluster à certains points de l'arborescence de rubriques. La portée de l'abonnement doit être définie pour autoriser le flux des abonnements de proxy vers d'autres gestionnaires de files d'attente du cluster. Vous devez restreindre la portée de l'abonnement d'une rubrique en cluster uniquement lorsque vous avez besoin de contrôler avec une granularité fine des rubriques spécifiques sur certains gestionnaires de files d'attente.

Lorsque vous utilisez une hiérarchie de publication/abonnement, la portée des abonnements est généralement contrôlée par cet attribut, combiné avec l'attribut [Portée de la publication](#).

L'attribut de rubrique **SUBSCOPE** est utilisé pour déterminer la portée des abonnements à une rubrique spécifique. Vous pouvez définir cet attribut sur l'une des valeurs suivantes :

### QMGR

Un abonnement reçoit uniquement les publications locales, et les abonnements proxy ne sont pas propagés aux gestionnaires de files d'attente éloignées.

### TOUT

Un abonnement de proxy est propagé vers les gestionnaires de files d'attente éloignées dans un cluster ou une hiérarchie de publication/abonnement, et l'abonné reçoit des publications locales et éloignées.

### ASPARENT

Utilisez le paramètre **SUBSCOPE** de la rubrique parent dans l'arborescence des rubriques.

Lorsque la portée de l'abonnement d'une rubrique est définie sur TOUT, directement ou via ASPARENT, vous pouvez limiter la portée des abonnements individuels à cette rubrique à QMGR en spécifiant MQSO\_SCOPE\_QMGR lors de la création de l'abonnement. Un abonnement à une rubrique dotée de la portée QMGR ne peut pas élargir la portée à ALL.

### Concepts associés

«Objets de rubrique d'administration», à la page 81

A l'aide d'un objet de rubrique d'administration, vous pouvez affecter des attributs spécifiques et autres que ceux par défaut aux rubriques.

### Tâches associées

[Configuration des réseaux de publication/abonnement réparti](#)

## Espaces de sujet

Un espace de sujet est l'ensemble de rubriques auquel vous pouvez vous abonner et que vous pouvez publier. Un gestionnaire de files d'attente dans une topologie de publication/abonnement réparti a un espace de sujet qui inclut potentiellement des rubriques dotées d'abonnements et publiées sur les gestionnaires de files d'attente connectés de cette topologie.

**Remarque :** Pour avoir une présentation des rubriques d'un gestionnaire de files d'attente, comme les objets de rubrique d'administration, les chaînes de rubrique et les arborescences de rubriques, voir «Rubriques», à la page 72. Dans cet article, les autres références aux *rubriques* se rapportent aux *chaînes de rubrique* sauf indication contraire.

Les rubriques sont initialement créées de l'une des manières suivantes :

- de façon administrative, lorsque vous définissez un objet de rubrique ou un abonnement durable,

- dynamiquement, lorsqu'une application crée une publication ou un abonnement de manière dynamique à une nouvelle rubrique.

Les rubriques sont propagées à d'autres gestionnaires de files d'attente via les abonnements de proxy ou en créant des objets de rubrique de cluster administratifs. Des abonnements de proxy se traduisent par le réacheminement des publications depuis le gestionnaire de files d'attente auquel un diffuseur de publications est connecté vers les gestionnaires de files d'attente des abonnés.

Les abonnements de proxy sont propagés entre tous les gestionnaires de files d'attente qui sont connectés les uns aux autres par des relations parent-enfant dans une hiérarchie de gestionnaires de files d'attente. Par conséquent, vous pouvez vous abonner sur un gestionnaire à une rubrique définie sur un autre gestionnaire de files d'attente de la hiérarchie. Tant qu'il existe un chemin connecté entre les gestionnaires de files d'attente, la manière dont ceux-ci sont connectés n'a pas d'importance.

Les abonnements de proxy sont également propagés aux abonnements à des rubriques de cluster dans un cluster de publication/abonnement. Une rubrique de cluster est une rubrique qui est connectée à un objet de rubrique ayant l'attribut **CLUSTER** ou qui hérite de l'attribut de son parent. Les rubriques qui sont pas des rubriques de cluster sont appelées rubriques locales et ne sont pas répliquées vers le cluster. Aucun abonnement de proxy n'est propagé vers le cluster depuis des abonnements à des rubriques locales.

Pour résumer, des abonnements de proxy sont créés pour des abonnés dans deux situations.

1. Un gestionnaire de files d'attente est membre d'une hiérarchie, et un abonnement de proxy est réacheminé vers le parent et les enfants du gestionnaire de files d'attente.
2. Un gestionnaire de files d'attente est membre d'un cluster, et la chaîne de rubrique d'abonnement est résolue en une rubrique qui est associée à un objet de rubrique de cluster. Lorsqu'une rubrique est une rubrique de cluster à *routage direct*, les abonnements de proxy sont transmis à tous les membres du cluster. Lorsque la rubrique est une rubrique de cluster *routée via un hôte de rubrique*, les abonnements de proxy sont transmis uniquement aux gestionnaires de files d'attente du cluster qui ont défini l'objet de rubrique en cluster. Pour plus d'informations, voir [«Clusters de publication/abonnement»](#), à la page 98.

Si un gestionnaire de files d'attente est un membre d'un cluster et d'une hiérarchie, les abonnements de proxy sont propagés par les deux mécanismes sans distribution de publications en double à l'abonné.

Les espaces de sujet de trois topologies de publication/abonnement sont décrits dans la liste suivante :

- [«Cas 1. Clusters de publication/abonnement»](#), à la page 112.
- [«Cas 2. Hiérarchies de publication / abonnement dans la version 7 ou les versions ultérieures»](#), à la page 113.
- [«Cas 3 Hiérarchies de publication / d'abonnement et flux dans IBM WebSphere MQ 6»](#), à la page 113.

Dans des rubriques distinctes, les tâches de configuration suivantes expliquent comment combiner des espaces de sujet.

- [Création d'un espace de sujet dans un cluster de publication/abonnement.](#)
- [Ajout d'un gestionnaire de files d'attente IBM WebSphere MQ 7 ou versions ultérieures dans des espaces de sujet IBM WebSphere MQ 6.](#)
- [Combinaison des espaces de sujet de plusieurs clusters.](#)
- [Combinaison et isolement des espaces de sujet dans plusieurs clusters.](#)
- [Publication et abonnement à des espaces de sujet dans plusieurs clusters.](#)

### **Cas 1. Clusters de publication/abonnement**

Dans cet exemple, supposons que le gestionnaire de files d'attente n'est pas connecté à une hiérarchie de publication/abonnement.

Si un gestionnaire de files d'attente est membre d'un cluster de publication/abonnement, son espace de sujet est constitué de rubriques locales et de rubriques de cluster. Les rubriques locales sont associées à des objets de rubrique sans l'attribut **CLUSTER**. Si un gestionnaire de files d'attente comporte des

définitions d'objet de rubrique local, son espace de sujet est différent de celui d'un autre gestionnaire de files d'attente du cluster qui possède également ses propres objets de rubrique définis localement.

Dans un cluster de publication/abonnement, vous ne pouvez pas vous abonner à une rubrique définie sur un autre gestionnaire de files d'attente, sauf si la rubrique à laquelle vous vous abonnez est résolue en un objet de rubrique de cluster.

Lorsque les mêmes définitions nommées d'un objet de rubrique en cluster sont requises sur plusieurs gestionnaires de files d'attente, par exemple lors de l'utilisation du *routage via un hôte de rubrique*, il est essentiel que toutes les définitions correspondent là où cela est nécessaire. Pour plus d'informations, voir [Création d'un espace de sujet dans un cluster de publication/abonnement](#).

Une définition locale d'un objet de rubrique, que cette définition soit pour une rubrique de cluster ou une rubrique locale, est prioritaire sur le même objet de rubrique défini ailleurs dans le cluster. La rubrique définie localement est utilisée, même si l'objet défini ailleurs est plus récent.

Il est important qu'un objet de rubrique de cluster soit associé à la même chaîne de rubrique partout dans le cluster. Vous ne pouvez pas modifier la chaîne de rubrique avec laquelle un objet de rubrique est associé. Pour associer le même objet de rubrique à une autre chaîne de rubrique, vous devez le supprimer et le recréer avec la nouvelle chaîne de rubrique. Si la rubrique est mise en cluster, cela revient à supprimer les copies de l'objet de rubrique stockées sur les autres membres du cluster, puis à créer des copies du nouvel objet de rubrique partout dans le cluster. Les copies de l'objet de rubrique font toutes référence à la même chaîne de rubrique.

Il est possible de créer par accident deux définitions du même objet de rubrique nommé sur deux gestionnaires de files d'attente différents, avec des chaînes de rubriques différentes. Cela peut être source de confusion car plusieurs définitions du même objet de rubrique avec des chaînes de rubrique différentes peuvent entraîner des résultats différents selon la manière dont la rubrique est référencée et à quel emplacement. Pour plus d'informations, voir [Plusieurs définitions de rubrique de cluster du même nom](#).

## **Cas 2. Hiérarchies de publication / abonnement dans la version 7 ou les versions ultérieures**

Dans cet exemple, supposons que le gestionnaire de files d'attente n'est pas membre d'un cluster de publication/abonnement.

Dans IBM WebSphere MQ 7 ou versions ultérieures, si un gestionnaire de files d'attente est membre d'une hiérarchie de publication/abonnement, son espace de sujet comprend toutes les rubriques définies localement et sur les gestionnaires de files d'attente connectés. L'espace de sujet de tous les gestionnaires de files d'attente d'une hiérarchie est le même. Il n'existe pas de séparation des rubriques en rubriques locales et en rubriques globales.

Définissez l'une des options **PUBSCOPE** et **SUBSCOPE** sur QMGR pour empêcher une publication sur une rubrique d'être transmises depuis un diffuseur de publications vers différents gestionnaires de files d'attente dans la hiérarchie.

Supposons que vous définissez un objet de rubrique A1abama avec la chaîne de rubrique USA/A1abama sur le gestionnaire de files d'attente QMA. Le résultat est le suivant :

1. L'espace de sujet dans QMA inclut désormais l'objet de rubrique A1abama et la chaîne de rubrique USA/A1abama.
2. Une application ou un administrateur peut créer un abonnement à l'adresse QMA à l'aide du nom d'objet de rubrique A1abama.
3. Une application peut créer un abonnement à n'importe quel sujet, y compris USA/A1abama, dans n'importe quel gestionnaire de files d'attente de la hiérarchie. Si QMA n'a pas été défini localement, la rubrique USA/A1abama est résolue par l'objet de rubrique SYSTEM.BASE.TOPIC.

## **Cas 3 Hiérarchies de publication / d'abonnement et flux dans IBM WebSphere MQ 6**

Avant IBM WebSphere MQ 7, l'espace de sujet était divisé en flux distincts, qui incluaient le flux par défaut présent sur tous les gestionnaires de files d'attente. Les publications ne peuvent pas être transférées entre les différents flux. Si des flux nommés sont utilisés, les espaces de sujet sur différents

gestionnaires de files d'attente peuvent être différents. Les rubriques sont séparées dans des rubriques dans le flux par défaut et des rubriques dans différents flux nommés.

**Remarque :** Chaque flux nommé constitue un espace de sujet distinct. Pour former une topologie connectée, chaque flux nommé doit exister sur les gestionnaires de files d'attente connectés. Supposons que le flux X est défini sur QMA et QMC, mais pas sur QMB. Si QMA est le parent de QMB, et si QMB est le parent de QMC, aucune rubrique du flux X ne peut être transférée entre QMA et QMC.

La définition des deux options **PUBSCOPE** et **SUBSCOPE** sur QMGR ou sur ALL nécessite un diffuseur de publications et un abonné à une rubrique pour échanger uniquement des publications pour une utilisation locale, ou uniquement des publications pour une utilisation globale.

Depuis IBM WebSphere MQ 7, aucun flux n'est disponible via l'API de publication/abonnement. Si vous utilisez la publication/l'abonnement en file d'attente sur un gestionnaire de files d'attente IBM WebSphere MQ 7, les flux sont mappés à différents objets de rubrique qui peuvent simuler l'effet des flux. Un flux est simulé en créant un objet de rubrique tenant lieu de rubrique racine pour toutes les rubriques du flux. Le gestionnaire de files d'attente mappe les publications et les abonnements entre le flux et la rubrique racine correspondante de chaque arborescence.

### Concepts associés

«Portée de la publication», à la page 110

Lorsque vous configurez un cluster ou une hiérarchie de publication/abonnement, la portée d'une publication détermine également si les gestionnaires de files d'attente transmettent une publication aux gestionnaires de files d'attente éloignées. Utilisez l'attribut de rubrique **PUBSCOPE** pour administrer la portée des publications.

«Portée de l'abonnement», à la page 111

La portée d'un abonnement contrôle si un abonnement sur un gestionnaire de files d'attente reçoit des publications qui sont publiées sur un autre gestionnaire de files d'attente dans un cluster ou une hiérarchie de publication/abonnement, ou s'il reçoit uniquement les publications de diffuseurs de publications locaux.

### Tâches associées

Configuration des réseaux de publication/abonnement réparti

## Multidiffusion IBM MQ

---

IBM MQ Multicast fournit une messagerie multidiffusion fiable, à haute distribution et à faible latence.

La multidiffusion est une forme de messagerie de type publication/abonnement efficace car elle peut contenir un nombre élevé d'abonnés sans que cela n'ait un impact négatif sur les performances. IBM MQ permet d'utiliser une messagerie multidiffusion fiable en utilisant des accusés de réception, des accusés réception négatifs et des numéros de séquence pour disposer d'une messagerie à haute distribution et faible latence.

La distribution fiable IBM MQ permet de distribuer les messages de manière quasiment simultanément pour qu'aucun destinataire n'ait un avantage. Comme la multidiffusion IBM MQ utilise le réseau pour distribuer les messages, aucun moteur de publication/abonnement n'est nécessaire pour distribuer les données. Une fois qu'une rubrique est mappée sur un adresse de groupe, le gestionnaire de files d'attente n'est pas requis, car les diffuseurs de publications et les abonnés fonctionnent en mode d'égal à égal. Ceci permet la réduction de la charge sur les serveurs de gestionnaire de files d'attente ; le gestionnaire ne représente plus une source d'incident potentielle.

### Concepts multidiffusion initiaux

La multidiffusion IBM MQ peut facilement s'intégrer dans les systèmes et les applications existants à l'aide de l'objet Informations de communication (COMMINFO). Deux zones d'objet TOPIC activent la configuration rapide des objets TOPIC existants pour prendre en charge ou pour ignorer le trafic multidiffusion.

## Objets requis pour la multidiffusion

Les informations suivantes sont une brève présentation des deux objets requis pour la multidiffusion IBM MQ :

### Objet **COMMINFO**

L'objet **COMMINFO** contient les attributs associés à une transmission multidiffusion. Pour plus d'informations sur les paramètres de l'objet **COMMINFO**, voir [DEFINE COMMINFO](#).

L'unique zone **COMMINFO** qui **DOIT** être défini est le nom de l'objet **COMMINFO**. Ce nom est ensuite utilisé pour identifier l'objet **COMMINFO** par rapport à une rubrique. La zone **GRPADDR** de l'objet **COMMINFO** doit être vérifié afin de s'assurer que la valeur est une adresse de groupe multidiffusion valide.

### Objet **TOPIC**

Une rubrique est l'objet des informations publiées dans un message de publication/abonnement, et une rubrique se définit en créant un objet **TOPIC**. Pour plus d'informations sur les paramètres de l'objet **TOPIC**, voir [DEFINE TOPIC](#).

Les rubriques existantes peuvent être utilisées avec la multidiffusion en modifiant les valeurs des paramètres de l'objet **TOPIC** suivants : **COMMINFO** et **MCAST**.

- **COMMINFO** Ce paramètre indique le nom de l'objet d'information de communication multidiffusion.
- **MCAST** Ce paramètre indique si la multidiffusion est autorisée à ce niveau dans l'arborescence des rubriques. Par défaut, **MCAST** est défini sur **ASPARENT**, ce qui signifie que l'attribut de multidiffusion de la rubrique est hérité du parent. Définir **MCAST** sur **ENABLED** autorise le trafic multidiffusion sur ce noeud.

## Rubriques et réseaux de multidiffusion

Les informations suivantes sont une présentation de ce qui se passe avec les abonnements avec des définitions d'abonnement et de rubrique différentes. Ces exemples supposent tous que le paramètre **COMMINFO** de l'objet **TOPIC** est défini sur le nom d'un objet **COMMINFO** valide :

### Activation de la rubrique définie sur la multidiffusion

Si le paramètre **MCAST** de la chaîne de rubrique est défini sur **ENABLED**, les abonnements des clients capables de multidiffusion sont autorisés et un abonnement de multidiffusion est réalisé sauf :

- S'il s'agit d'un abonnement durable d'un client capable de multidiffusion.
- S'il s'agit d'un abonnement non géré d'un client capable de multidiffusion.
- S'il s'agit d'un abonnement d'un client non capable de multidiffusion.

Dans ces cas, un abonnement sans multidiffusion est défini et les abonnements sont rétrogradés vers la publication/l'abonnement normal.

### Désactivation de la rubrique définie sur la multidiffusion

Si le paramètre **MCAST** de la chaîne de rubrique est défini sur **DISABLED**, un abonnement non multidiffusion est toujours réalisé et les abonnements sont rétrogradés vers la publication/l'abonnement normal.

### Rubrique défini sur la multidiffusion uniquement

Si le paramètre **MCAST** de la chaîne de rubrique est défini sur **ONLY**, les abonnements des clients capables de multidiffusion sont autorisés et un abonnement multidiffusion est réalisé sauf :

- Il s'agit d'un abonnement durable : les abonnements durables sont rejetés avec le code anomalie [2436 \(0984\) \(RC2436\): MQRC\\_DURABILITY\\_NOT\\_ALLOWED](#)
- Il s'agit d'un abonnement non géré : les abonnements non gérés sont rejetés avec le code anomalie [2046 \(07FE\) \(RC2046\): MQRC\\_OPTIONS\\_ERROR](#)
- Il s'agit d'un abonnement d'un client ne pouvant pas prendre en charge la multidiffusion : ces abonnements sont rejetés avec le code anomalie [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#)
- Il s'agit d'un abonnement d'une application locale : ces abonnements sont rejetés avec le code anomalie [2560 \(0A00\) \(RC2560\): MQRC\\_MULTICAST\\_ONLY](#)

MQ Telemetry comprend un service de télémétrie (MQXR) qui fait partie d'un gestionnaire de files d'attente ainsi que de clients de télémétrie que vous pouvez écrire vous-même ou télécharger gratuitement, ainsi que des interfaces d'administration de l'explorateur ou de ligne de commande. La télémétrie représente l'administration et la collecte de données à partir d'un large éventail de périphériques distants. MQ Telemetry permet d'intégrer la collection de données et de contrôler les périphériques dotés d'applications Web.

Depuis IBM WebSphere MQ 7.1, MQ Telemetry est un composant d'IBM MQ. La mise à niveau de ces versions consiste essentiellement à installer la dernière version d'IBM MQ.

Toutefois, depuis IBM MQ 8.0, il n'est plus livré avec le produit. Des exemples d'application similaires restent disponibles gratuitement depuis Eclipse Paho et [MQTT.org](http://MQTT.org). Voir [IBM MQ Telemetry Transport sample programs](#).

MQ Telemetry étant un composant d'IBM WebSphere MQ 7.1 et des versions ultérieures, MQ Telemetry peut être installé avec le produit principal ou après l'installation du produit principal. Pour des informations sur la migration, voir [Migration de MQ Telemetry sous Windows](#) et [Migration de MQ Telemetry sous Linux](#).

Les composants suivants sont inclus dans MQ Telemetry :

### Canaux de télémétrie

Utilisez les canaux de télémétrie pour gérer la connexion des clients MQTT à IBM MQ . Les canaux de télémétrie utilisent de nouveaux objets IBM MQ , tels que SYSTEM.MQTT . TRANSMIT . QUEUE, pour interagir avec IBM MQ.

### Service de télémétrie (MQXR)

Les clients MQTT utilisent le service de télémétrie SYSTEM.MQXR . SERVICE pour se connecter aux canaux de télémétrie.

### Prise en charge d'IBM MQ Explorer pour MQ Telemetry

MQ Telemetry peut être administré à l'aide de IBM MQ Explorer.

### Documentation

La documentation MQ Telemetry est incluse dans la documentation du produit IBM MQ à partir de la IBM WebSphere MQ 7.1. La documentation relative au kit de développement des clients Java et C est fournie dans la documentation du produit, sous la forme Javadoc et dans le format HTML.

## Concepts de télémétrie

Vous collectez des informations à partir de l'environnement qui vous entoure et décidez les actions à entreprendre. En tant que consommateur, vous vérifiez ce que vous avez en stock avant d'acheter votre nourriture. Vous voulez savoir la durée de votre voyage si vous partez maintenant, avant d'effectuer votre réservation. Vous vérifiez les symptômes avant de vous rendre chez le médecin. Vous vérifiez l'horaire du bus et pouvez ainsi décider d'attendre ou non. Les informations qui vous aident à prendre ces décisions sont fournies par des compteurs et des appareils, un mot écrit sur une feuille de papier ou sur un écran et par vous-même. Où que vous soyez, et quand vous le voulez, vous collectez des informations, les rassemblez, les analysez et les utilisez.

Si les sources d'information sont trop éparses ou inaccessibles, il est difficile et coûteux de retrouver les informations les plus appropriées. Si vous voulez effectuer beaucoup de modifications ou s'il est difficile d'effectuer ces modifications, elles ne seront pas effectuées ou le seront au moment le moins opportun.

Que penseriez-vous si le coût de la collecte d'informations et le contrôle d'unités largement disséminées était réduit de manière significative par la connexion de ces appareils à Internet via la technologie numérique ? Ces informations pourraient alors être analysées à l'aide des ressources d'Internet et des entreprises. Vous auriez alors de nombreuses opportunités de prendre des décisions et d'agir en connaissance de cause.

Les tendances technologiques, les pressions environnementales et économiques permettent d'effectuer ces changements :

1. Le coût de la connexion et du contrôle des détecteurs et des actionneurs diminue en raison de la standardisation et de la connexion à des processeurs numériques à faible coût.
2. Internet et les technologies liées à Internet sont de plus en plus utilisés pour la connexion des appareils. Dans certains pays, le nombre de connexions aux applications Internet réalisées via les téléphones mobiles dépasse celui effectué avec des ordinateurs portables. D'autres types d'unités suivront la probablement la même tendance.
3. Internet et les technologies liées à Internet facilitent l'accès aux données pour une application. Un accès plus facile aux données permet d'utiliser l'analyse de données pour transformer les données des détecteurs en informations utiles pour plusieurs solutions.
4. L'utilisation intelligente des ressources est souvent un moyen moins cher et plus rapide de réduire les coûts et les émissions de carbone. Les alternatives sont les suivantes : trouver de nouvelles ressources ou développer de nouvelles technologies pour utiliser les ressources existantes peut être une solution à long terme. A court terme, le développement de nouvelles technologies ou la recherche de nouvelles ressources est souvent plus risqué, plus lent et plus coûteux que l'amélioration des solutions existantes.

### Exemple

Un exemple illustre comment ces tendances créent de nouvelles opportunités pour interagir intelligemment avec l'environnement.

La convention internationale SOLAS (Safety of Life at Sea) nécessite le déploiement du Système d'identification automatique (SIA) sur de nombreux navires. Ce système est obligatoire sur les navires marchands de plus de 300 tonnes et les navires de passagers. Le système SIA est principalement un système permettant d'éviter les collisions lors du cabotage. Il est utilisé par les autorités maritimes pour surveiller et contrôler les eaux côtières.

Des passionnés dans le monde entier déploient des stations de suivi SIA et envoient les informations relatives à la navigation de cabotage sur Internet. D'autres passionnés écrivent des applications combinant des informations provenant du système SIA avec d'autres informations provenant d'Internet. Les résultats sont mis sur les sites Web et publiés via Twitter ou l'envoi de SMS.

Dans une application, les informations provenant de stations SIA près de Southampton sont combinées avec celles des propriétaires de navires et des informations géographiques. L'application fournit des informations en direct sur les arrivées et les départs de bateaux sur Twitter. Les voyageurs réguliers utilisant les bateaux entre Southampton et l'île de Wight s'abonnent à la distribution de nouvelles en utilisant Twitter ou les SMS. Si le fil indique que le bateau est en retard, les voyageurs peuvent retarder leur départ et prendre le bateau lorsqu'il accoste plus tard que son heure d'arrivée prévue.

Pour plus d'exemples, voir [«Scénarios d'utilisation de la télémétrie»](#), à la page 119.

### Tâches associées

[Installation de MQ Telemetry](#)

[Administration de MQ Telemetry](#)

[Migration d'MQ Telemetry sous Windows](#)

[Migration d'MQ Telemetry sous Linux](#)

[Développement d'applications pour MQ Telemetry](#)

[Traitement des incidents MQ Telemetry](#)

### Référence associée

[Référence MQ Telemetry](#)

Windows

Linux

AIX

## Présentation de MQ Telemetry

Les personnes, les commerces et les gouvernements veulent de plus en plus utiliser MQ Telemetry pour interagir plus intelligemment avec l'environnement dans lequel nous vivons et nous travaillons. MQ Telemetry permet de connecter toutes sortes d'unités à Internet et aux entreprises, et de réduire les coûts de création d'applications pour les unités intelligentes.

## Qu'est-ce que MQ Telemetry ?

- Il s'agit d'une fonction d'IBM MQ qui étend le réseau principal de la messagerie universelle fournie par IBM MQ à toute une gamme de détecteurs distants, de mécanismes d'accès et d'appareils de télémétrie. MQ Telemetry étend IBM MQ de sorte qu'il puisse connecter des applications intelligentes d'entreprises, des services et des décisionnaires aux réseaux d'appareils instrumentés.
- Les principaux composants de MQ Telemetry sont les suivants :

### **Le service MQ Telemetry (MQXR)**

Ce service s'exécute dans le serveur IBM MQ et utilise le protocole IBM MQ Telemetry Transport (MQTT) pour communiquer avec les périphériques de télémétrie.

### **Les applications MQTT que vous écrivez**

Ces applications contrôlent les informations transmises entre les dispositifs de télémétrie et le gestionnaire de files d'attente IBM MQ, ainsi que les actions effectuées en réponse à ces informations. Pour faciliter la création de ces applications, vous utilisez des bibliothèques client MQTT.

## Intérêt

- MQTT est un transport de messagerie qui permet de créer des implémentations MQTT pour un large éventail de périphériques.
- Les clients MQTT peuvent s'exécuter sur des périphériques plus petits ayant des ressources limitées.
- MQTT fonctionne efficacement sur les réseaux à faible bande passante, où le coût d'envoi des données est élevé ou peut être fragile.
- La distribution des messages est assurée et découplée de l'application.
- Les programmeurs d'application n'ont pas besoin d'avoir des connaissances en programmation des communications.
- Des messages peuvent être échangés avec d'autres applications de messagerie. Ces applications peuvent être une autre application de télémétrie, une interface MQI, JMS ou une application de messagerie d'entreprise.

## Utilisation

- Des exemples de scripts sont fournis avec un exemple d'application client IBM MQ Telemetry Transport v3 (mqttv3app.jar). Voir [Programmes exemples IBM MQ Telemetry Transport](#).
- Utilisez IBM MQ Explorer et les outils associés pour administrer la fonction de télémétrie d'IBM MQ.
- Utilisez les bibliothèques client pour créer des applications MQTT qui connectent un gestionnaire de files d'attente et qui utilisent la messagerie de publication/abonnement.
- Distribuez l'application et la bibliothèque client sur le dispositif sur lequel l'application doit être exécutée.

## Comment cela fonctionne-t-il ?

- MQTT est un protocole de publication/abonnement. Une application client MQTT peut publier des messages vers un serveur MQTT ou s'abonner aux messages envoyés par les applications qui se connectent à un serveur MQTT.
- Les applications client MQTT utilisent les bibliothèques client qui implémentent le transport des messages MQTT.
- Une application client MQTT de base fonctionne comme un client MQ standard, mais elle peut s'exécuter sur un très large éventail de plateformes et de réseaux.
- Le service MQ Telemetry (MQXR) transforme un gestionnaire de files d'attente IBM MQ en serveur MQTT.

- Lorsqu'un gestionnaire de files d'attente IBM MQ fait office de serveur MQTT, les autres applications qui se connectent au gestionnaire de files d'attente peuvent s'abonner aux messages du client MQTT et les recevoir.
- Le gestionnaire de files d'attente agit comme un routeur qui distribue des messages des applications de publications aux applications d'abonnement.
- Les messages peuvent être distribués entre différents types d'applications client. Par exemple, entre les clients Telemetry et les clients JMS.

**Remarque :** MQ Telemetry remplace les noeuds SCADA qui ont été retirés de la version 7 de WebSphere Message Broker (également appelée IBM Integration Bus) et s'exécute sous Windows, Linux et AIX.

Windows

Linux

AIX

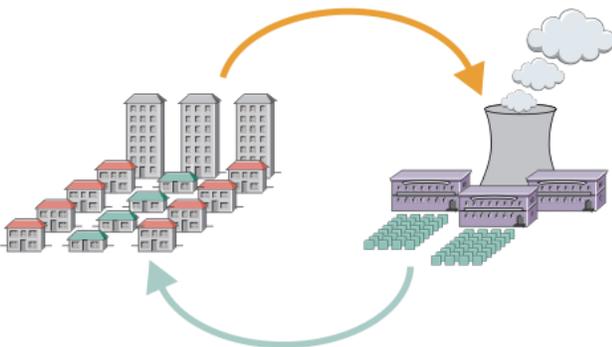
## Scénarios d'utilisation de la télémétrie

La télémétrie est la détection automatisée, la mesure des données et le contrôle des appareils à distance. L'accent est mis sur la transmission des données à partir des unités vers un point de contrôle central. La télémétrie comprend également l'envoi d'informations de configuration et de contrôle aux unités.

MQ Telemetry permet de connecter de petits périphériques utilisant le MQTT protocol et de connecter ces périphériques à d'autres applications à l'aide d'IBM MQ. MQ Telemetry comble le fossé qui existait entre les unités et Internet facilitant la création de "solutions intelligentes". Ces solutions permettent d'utiliser l'abondance d'informations disponibles sur Internet et dans les applications d'entreprise pour les mettre à disposition d'applications qui surveillent et contrôlent les unités.

Les diagrammes suivants illustrent quelques utilisations typiques de MQ Telemetry :

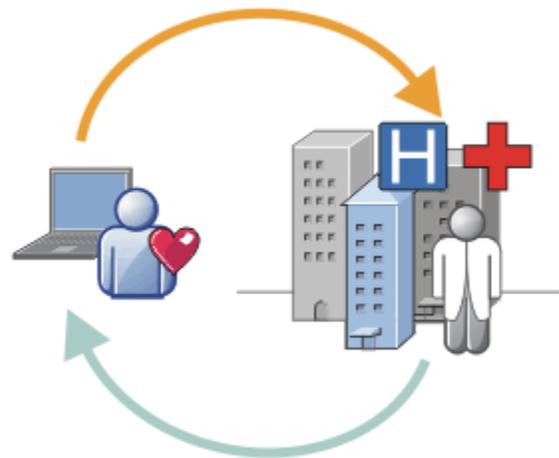
### Télémétrie : Electricité intelligente

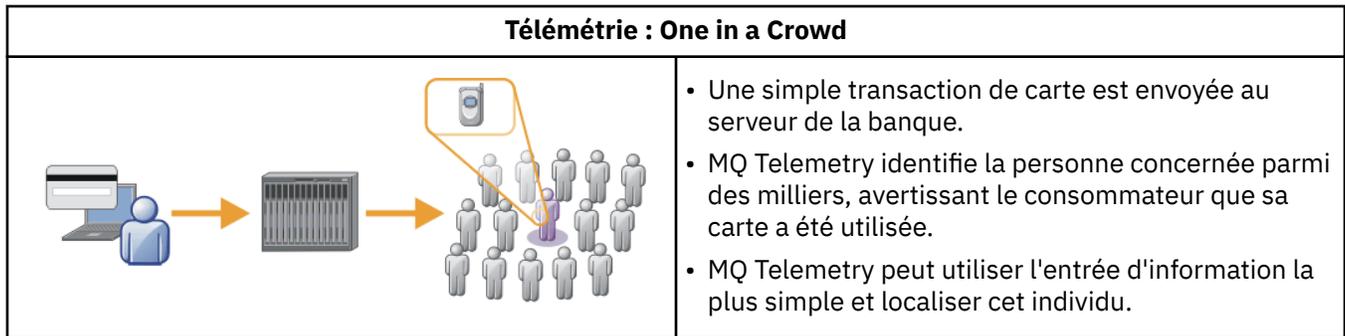


- Messages MQTT contenant les données relatives à la consommation d'énergie envoyées au fournisseur de services.
- MQ Telemetry envoie des commandes de contrôle CONTROL COMMANDS basées sur l'analyse des données relatives à la consommation d'énergie.
- Pour plus d'informations, voir le cas d'utilisation suivant : «Scénarios d'utilisation de la télémétrie : Surveillance et contrôle de l'énergie à domicile», à la page 122

### Télémétrie : Services de santé intelligent

- MQ Telemetry envoie les données de santé à votre hôpital et à votre médecin.
- Des alertes de message MQTT peuvent être envoyées en fonction de l'analyse des données de santé.
- Pour plus d'informations, voir le cas d'utilisation suivant : «Scénarios d'utilisation de la télémétrie : Surveillance de patient à domicile», à la page 120





Les scénarios d'utilisation décrits dans les sous-rubriques sont tirés d'exemples réels. Ils fournissent quelques exemples d'utilisation de la télémétrie et abordent les problèmes communs que la technologie devra résoudre.

Windows

Linux

AIX

## Scénarios d'utilisation de la télémétrie :

### Surveillance de patient à domicile

Dans le cadre de la collaboration entre IBM et un établissement de soins pour un système de surveillance d'un patient atteint de maladie cardiaque, un défibrillateur cardiovertteur communique avec un hôpital. Les données du patient et de l'appareil implanté sont transférées à l'aide de la télémétrie de fréquence radio au périphérique MQTT au domicile d'un patient.

Le transfert a lieu généralement la nuit vers un émetteur situé à proximité du lit du patient. Cet émetteur transmet les données de manière sécurisée via le système téléphonique à l'hôpital où les données sont analysées.

Le système réduit le nombre de visites que le patient doit effectuer chez le médecin. Il permet de détecter lorsque le patient ou l'appareil doit être contrôlé et, en cas d'urgence, il alerte le médecin de garde.

Les caractéristiques de la collaboration entre IBM et l'établissement de soins sont communes dans plusieurs scénarios d'utilisation de la télémétrie :

#### Invisibilité

L'appareil ne nécessite aucune intervention de l'utilisateur autre que l'alimentation électrique, une ligne téléphonique et l'obligation pour l'utilisateur de se trouver à proximité de l'appareil une partie de la journée. Son utilisation est simple et fiable.

Pour éviter au patient d'avoir à configurer l'appareil, ce dernier est livré déjà configuré. Le patient doit juste le brancher sur le secteur. Le fait que l'appareil ne soit pas configuré par le patient simplifie son utilisation et diminue le risque de configuration erronée.

Le client MQTT est incorporé au périphérique. Le développeur du périphérique incorpore l'implémentation du client MQTT dans l'appareil et le développeur, ou fournisseur, configure le client MQTT dans le cadre de la configuration.

Le client MQTT est fourni sous la forme d'un fichier JAR Java SE, que le développeur inclut dans son application Java. Pour les environnements non-Java, tels que celui-ci, le développeur du périphérique peut implémenter un client dans une langue différente en utilisant les formats et le protocole MQTT publiés. Le développeur peut également utiliser l'un des clients C fournis en tant que bibliothèques partagées pour les plateformes Windows, Linux et ARM.

#### Connectivité inégale

La communication entre le défibrillateur et l'hôpital se fait via un réseau dont les caractéristiques sont inégales. Deux réseaux différents sont utilisés pour résoudre les différents problèmes liés à la collecte de données du patient et à l'envoi de ces données à l'hôpital. Entre le patient et l'unité MQTT un réseau de fréquence radio de faible puissance et de portée limitée est utilisé. L'émetteur se connecte à l'hôpital en utilisant une connexion TCP/IP de réseau privé virtuel via une ligne téléphonique à faible bande passante.

Il n'est pas toujours pratique de trouver un moyen de connecter tous les appareils directement à un réseau IP. La solution habituelle consiste à utiliser deux réseaux connectés à un concentrateur. Le périphérique MQTT est un simple concentrateur qui stocke les informations du patient et les transfère ensuite à l'hôpital.

## Sécurité

Le médecin doit avoir la preuve de l'authenticité des données du patient et le patient doit pouvoir bénéficier du respect de la confidentialité des données le concernant.

Dans certaines situations, il suffit de chiffrer la connexion à l'aide du réseau privé virtuel ou de TLS. Dans d'autres, il est souhaitable de conserver les données en sécurité même après leur stockage.

Il arrive parfois que le dispositif de télémétrie ne soit pas sûr. Il peut se trouver dans un habitat partagé, par exemple. L'utilisateur du dispositif doit s'authentifier pour s'assurer que les données proviennent du patient approprié. Le dispositif lui-même peut être authentifié sur le serveur à l'aide de TLS, de même que le serveur s'authentifie sur le dispositif.

Le canal de télémétrie entre le dispositif et le gestionnaire de files d'attente prend en charge JAAS pour l'authentification utilisateur et TLS pour le chiffrement de la communication et l'authentification du dispositif. L'accès à une publication est contrôlé par le gestionnaire des droits d'accès aux objets dans IBM MQ.

L'identificateur utilisé pour authentifier l'utilisateur peut être mappé sur un identificateur différent, tel qu'une identité de patient commune. Un identificateur commun simplifie la configuration de l'autorisation d'accès aux rubriques de publication dans IBM MQ.

## Connectivité

La connexion entre le périphérique MQTT et l'hôpital s'effectue via une ligne commutée et fonctionne sur une bande passante dont le débit est de 300 bauds.

Pour fonctionner efficacement à 300 bauds, le MQTT protocol ajoute uniquement quelques octets à un message en plus des en-têtes TCP/IP.

Le MQTT protocol fournit une transmission unique de messagerie de type *autonome après diffusion*, qui permet de maintenir un faible temps de latence. Il peut également utiliser plusieurs transmissions pour garantir la distribution *au moins une fois et une seule fois* si la distribution garantie est plus importante que le temps de réponse. Pour garantir la distribution, les messages sont stockés sur le dispositif jusqu'à ce qu'ils soient distribués. Si un dispositif est relié via une connexion sans fil, la distribution garantie est particulièrement utile.

## Extensibilité

Les dispositifs de télémétrie sont normalement déployés en grande quantité, de dizaines de milliers à plusieurs millions d'unité.

La connexion de dispositifs sur un système entraîne des demandes importantes pour une solution. Les demandes sont d'ordre commercial, le coût des dispositifs et leur logiciel, et d'ordre administratif, la gestion des licences, des dispositifs et des utilisateurs. Les demandes d'ordre technique sont relatives à la charge sur le réseau et sur les serveurs.

L'ouverture des connexions utilise plus de ressources serveur que le maintien des connexions ouvertes. Mais dans un scénario tel que celui-ci, l'utilisation de lignes téléphoniques, le développement des connexions signifie que les connexions ne sont pas ouvertes plus longtemps que nécessaire. Les transferts de données sont largement effectués par lots. Les connexions peuvent être planifiées pendant la nuit de manière à éviter une augmentation soudaine des connexions à l'heure du coucher.

Au niveau du client, l'évolutivité des clients est favorisée par la configuration client minimale requise. Le client MQTT est incorporé au périphérique. Aucune étape de configuration ou d'acceptation de licence client MQTT ne doit être intégrée au déploiement des périphériques vers les patients.

Sur le serveur, MQ Telemetry a une cible initiale de 50 000 connexions ouvertes par gestionnaire de files d'attente.

Les connexions sont gérées à l'aide d'IBM MQ Explorer. IBM MQ Explorer filtre les connexions qui doivent être affichées pour un nombre gérable. Avec un schéma choisi d'identificateurs d'allocation aux clients, vous pouvez filtrer des connexions en fonction de l'emplacement géographique ou par ordre alphabétique de noms de patient.

Windows

Linux

AIX

## Scénarios d'utilisation de la télémétrie :

### Surveillance et contrôle de l'énergie à domicile

Les compteurs intelligents collectent plus d'informations sur la consommation d'énergie que les compteurs traditionnels.

Les compteurs intelligents sont souvent couplés à un réseau de télémétrie local pour surveiller et contrôler les appareils individuels dans une maison. Certains d'entre eux sont également connectés à distance pour la surveillance et le contrôle à distance.

La connexion à distance pourrait être configurée par un individu, une unité d'alimentation ou un point de contrôle central. Le point de contrôle à distance peut lire les données d'utilisation de l'alimentation et fournir des données d'utilisation. Il peut fournir des données pour influencer cette utilisation telles que la tarification continue et les informations météorologiques. Il peut limiter la charge pour améliorer l'efficacité globale de la production d'alimentation électrique.

Les compteurs intelligents commencent à se déployer à grande échelle. Au Royaume Uni, par exemple, le gouvernement est en pleine consultation pour l'installation de compteurs intelligents dans chaque foyer.

Les scénarios d'utilisation de compteurs à domicile ont plusieurs caractéristiques en commun :

#### Invisibilité

Le compteur ne doit nécessiter aucune intervention de l'utilisateur, sauf si ce dernier le souhaite pour économiser de l'énergie. Le compteur ne doit pas affecter la fiabilité de fourniture d'énergie aux appareils individuels.

Un client MQTT peut être intégré au logiciel déployé avec le compteur, et ne nécessite pas d'installation ou de configuration séparée.

#### Connectivité inégale

La communication entre les appareils et le compteur intelligent nécessite des normes différentes de connectivité plus importantes qu'entre le compteur et le point de connexion à distance.

La connexion entre le compteur intelligent et les appareils doit être hautement disponible et être conforme aux normes de réseau applicables à un réseau domestique.

Le réseau à distance est susceptible d'utiliser plusieurs connexions physiques. Certaines d'entre elles, comme les connexions cellulaires, ont un coût de transmission élevé et peuvent être intermittentes. La spécification MQTT version 3 est destinée aux connexions à distance et aux connexions entre les adaptateurs locaux et le compteur intelligent.

La connexion entre les prises d'alimentation et les appareils, et le compteur utilise un réseau domestique, tel que Zigbee. MQTT for sensor networks (MQTT-S), a été conçu pour fonctionner avec Zigbee et d'autres protocoles de réseau à faible bande passante. MQ Telemetry ne prend pas en charge directement MQTT-S. Il nécessite une passerelle pour connecter MQTT-S à MQTT v3.

Comme dans le cas de la surveillance de patient à domicile, les solutions pour la surveillance et le contrôle nécessitent plusieurs réseaux connectés en utilisant le compteur intelligent comme un concentrateur.

#### Sécurité

Il existe plusieurs problèmes de sécurité liés aux compteurs intelligents. Ces problèmes comprennent l'irréfutabilité des transactions, l'autorisation des actions de contrôle qui sont initiées et la confidentialité des données de consommation électrique.

Pour assurer la confidentialité des données, les données transférées entre le compteur et le point de contrôle à distance par MQTT peuvent être chiffrées avec TLS. Pour assurer l'autorisation des

actions de contrôle, la connexion MQTT entre le compteur et le point de contrôle à distance peut être mutuellement authentifiée à l'aide de TLS.

## Connectivité

La nature physique du réseau à distance peut varier considérablement. Le réseau peut utiliser une connexion à large bande existante ou un réseau mobile avec des coûts d'appel élevés et une disponibilité intermittente. En matière de coûts élevés et d'intermittence, les connexions MQTT constituent un protocole fiable et efficace ; voir «Scénarios d'utilisation de la télémétrie : Surveillance de patient à domicile», à la page 120.

## Extensibilité

Finalement, les sociétés de distribution d'électricité ou les points de contrôle centraux envisagent de déployer des dizaines de millions de compteurs intelligents. Initialement le nombre de compteurs prévu par déploiement se situait entre des dizaines et des centaines de milliers. Ce chiffre est comparable à la cible MQTT initiale de 50 000 connexions client ouvertes par gestionnaire de files d'attente.

Un aspect critique de l'architecture pour la surveillance et le contrôle de l'énergie à domicile est l'utilisation du compteur intelligent comme concentrateur de réseau. Chaque adaptateur d'appareil est un détecteur séparé. En les connectant à un concentrateur local utilisant MQTT, le concentrateur peut concentrer le flux de données sur une simple session TCP/IP avec le point de contrôle central, et peut également stocker les messages pendant une courte période pour surmonter l'indisponibilité de session.

Les connexions à distance doivent être laissées ouvertes dans les scénarios d'énergie à domicile pour deux raisons. Tout d'abord parce que l'ouverture des connexions peut prendre du temps lié à l'envoi des demandes. Le temps nécessaire à l'ouverture de nombreuses connexions pour envoyer des demandes de "limitation de charge" dans un intervalle court est trop long. Ensuite, pour recevoir des demandes de limitation de charge à partir de la société distributrice d'électricité, la connexion doit d'abord être ouverte par le client. Avec MQTT, les connexions sont toujours démarrées par le client, et pour recevoir les demandes de limitation de charge de la société distributrice d'électricité, la connexion doit restée ouverte.

Si le débit de l'ouverture de connexions est critique, ou que le serveur initie des demandes prioritaires, la solution consiste généralement à conserver plusieurs connexions ouvertes.

## Windows Linux AIX Scénarios d'utilisation de télémétrie : Identification par radio fréquences (RFID)

La technologie RFID consiste en l'utilisation d'une étiquette RFID pour identifier et faire le suivi d'un objet via une liaison sans fil. Les étiquettes RFID peuvent être lues par un grand nombre de compteurs et, ceci même si l'étiquette n'est pas dans l'axe de visée du lecteur RFID. Les étiquettes passives sont activées par un lecteur RFID. Les étiquettes actives émettent sans activation externe. Les étiquettes actives doivent disposer d'une alimentation. Les étiquettes passives peuvent incorporer une alimentation afin d'en accroître la plage.

La technologie RFID est utilisée dans de nombreuses applications et les types de scénario d'utilisation peuvent varier énormément. Les scénarios d'utilisation RFID, ainsi que les scénarios d'utilisation de la surveillance de patient à domicile, et du contrôle et de la surveillance énergétique à domicile comportent des similitudes et des différences.

## Invisibilité

Dans de nombreux scénarios d'utilisation, le lecteur RFID est déployé en grand nombre et doit fonctionner sans intervention de l'utilisateur. Le lecteur inclut un client MQTT intégré pour communiquer avec un point de contrôle central.

Par exemple, dans un entrepôt de distribution, un lecteur mobile pour détecter une palette. Le lecteur active l'étiquette RFID des articles se trouvant sur la palette et envoie les données et les demandes aux applications centrales. Les données sont utilisées pour la mise à jour de l'emplacement du stock. Les demandes contrôlent ensuite l'étape suivante pour la palette, comme par exemple son

déplacement vers une baie particulière. Les compagnies aériennes, et les systèmes de gestion des bagages dans les aéroports utilisent la technologie RFID de la sorte.

Dans certains cas d'utilisation RFID, le lecteur dispose d'un environnement informatique standard, tel que Java Platform, Micro Edition (Java ME). Dans ces cas, le client MQTT peut être déployé dans une étape de configuration distincte, après la fabrication.

### **Connectivité inégale**

Les lecteurs RFID doivent être séparés de l'unité de contrôle locale qui contient un client MQTT, ou chaque lecteur peut disposer d'un client MQTT intégré. Normalement, les facteurs de localisation géographique ou de communications indiquent le choix de la topologie.

### **Sécurité**

La confidentialité et l'authenticité sont des préoccupations de sécurité lors de l'utilisation des étiquettes RFID. Les étiquettes RFID sont peu visibles, et peuvent être surveillées secrètement, espionnées ou trafiquées.

La solution aux problèmes de sécurité RFID accroît l'opportunité de déployer de nouvelles solutions RFID. Bien que l'exposition à la sécurité se trouve au sein même de l'étiquette RFID, et dans le lecteur local, l'utilisation du traitement d'informations centrales suppose des approches permettant de contrer les différentes menaces. Par exemple, le trafic d'étiquette peut être détecté en mettant en corrélation de manière dynamique les niveaux de stock et les livraisons et répartitions.

### **Connectivité**

Les applications RFID comprenaient à la fois le stockage et la retransmission d'informations par lots collectées à partir des lecteurs RFID et les demandes immédiates. Dans le scénario d'entrepôt de distribution, le lecteur RFID est connecté en permanence. Lorsqu'une étiquette est lue, elle est publiée avec les informations relatives au lecteur. L'application de l'entrepôt renvoie la réponse au lecteur.

Dans l'application d'entrepôt, le réseau est normalement fiable et les demandes immédiates peuvent utiliser des messages *autonomes après diffusion* pour des temps d'attente faibles. Les données par lot en mode différé peuvent utiliser la messagerie *une seule fois* pour minimiser les coûts de gestion associés à la perte de données.

### **Extensibilité**

Si l'application RFID nécessite des réponses immédiates, de l'ordre d'une seconde ou deux, les lecteurs RFID doivent rester connectés.

## **Scénarios d'utilisation de la télémétrie : Détection d'environnement**

La détection d'environnement utilise la télémétrie pour collecter des informations sur le niveau et la qualité des eaux de rivière, les polluants atmosphériques et d'autres données environnementales.

Les détecteurs sont souvent localisés dans des lieux distants sans accès aux communications filaires. La bande passante sans fil est onéreuse et sa fiabilité peut être faible. En général, plusieurs détecteurs d'environnement dans une zone géographique réduite sont connectés à une unité de surveillance locale dans un emplacement sûr. Les connexions locales peuvent être filaires ou sans fil.

### **Invisibilité**

Les unités de détecteur risquent d'être moins accessibles, moins bien alimentées et déployées en plus grand nombre que l'unité de contrôle centrale. Les détecteurs sont parfois "muets", et le dispositif de surveillance locale inclut les adaptateurs pour transformer et stocker les données de détecteur. L'unité de surveillance est susceptible d'incorporer un ordinateur à usage général prenant en charge Java Platform, Standard Edition (Java SE) ou Java Platform, Micro Edition (Java ME). Il est peu probable que l'invisibilité soit une exigence majeure lors de la configuration du client MQTT.

## Connectivité inégale

Les fonctions de capteurs et le coût de la bande passante de la connexion à distance se traduit de manière classique par l'utilisation d'un concentrateur de surveillance local connecté au serveur central.

## Sécurité

La sécurité n'est pas une exigence majeure sauf si elle est utilisée dans un scénario d'utilisation militaire ou de défense.

## Connectivité

Plusieurs utilisateurs ne nécessitent pas la surveillance continue ou la disponibilité des données. Les données d'exception, telles que l'alerte en cas d'inondation, doivent être transférées immédiatement. Les données de détecteur sont agrégées sur le moniteur local pour réduire les coûts de connexion et de communication, puis elles sont transférées à l'aide de connexions planifiées. Les données d'exception sont transférées dès qu'elles sont détectées au niveau du moniteur.

## Extensibilité

Les détecteurs sont concentrés autour des concentrateurs locaux et les données de détecteur sont agrégées en paquets qui sont transmis suivant un planning. Ces deux facteurs réduisent la charge sur le serveur central qui serait imposée par l'utilisation directe de détecteurs connectés.

## Scénarios d'utilisation de la télémétrie :

### Applications mobiles

Les applications mobiles sont des applications qui s'exécutent sur des dispositifs sans fil. Ces dispositifs sont soit des plateformes d'application génériques, soit des dispositifs personnalisés.

Les plateformes habituelles comprennent des ordinateurs de poche tels que les téléphones et les assistants de données personnels et les appareils portables tels que les ordinateurs bloc-notes. Les unités personnalisées utilisent un matériel spécialement adapté à des applications spécifiques. Un dispositif permettant de livrer un colis contre signature est un exemple de dispositif mobile. Les applications se trouvant sur les dispositifs mobiles sont souvent conçues sur une plateforme logicielle générique.

### Invisibilité

Le déploiement d'applications mobiles personnalisées est géré et peut inclure la configuration d'application client MQTT. Il est peu probable que l'invisibilité soit une exigence majeure lors de la configuration du client MQTT.

### Connectivité inégale

Contrairement à la topologie de concentrateur local des scénarios précédents, les clients mobiles se connectent à distance. La couche d'application client se connecte directement à une application du concentrateur central.

### Sécurité

Avec peu de sécurité physique, l'unité mobile et l'utilisateur mobile doivent être authentifiés. TLS est utilisé pour confirmer l'identité de l'unité, et JAAS pour authentifier l'utilisateur.

### Connectivité

Si l'application mobile dépend de la couverture sans fil, elle doit pouvoir fonctionner hors ligne et faire face efficacement à une interruption de la connexion. Dans cet environnement, l'objectif est de rester connecté, mais l'application doit pouvoir stocker et transférer les messages. Les messages, qui sont souvent des commandes ou des confirmations de distribution, ont une importante valeur commerciale. Ils doivent être stockés et transférés de manière fiable.

## Extensibilité

L'évolutivité n'est pas un élément majeur. Le nombre de clients d'application n'est pas censé dépasser des milliers ou des dizaines de milliers d'unités dans les scénarios d'application mobile.

Windows

Linux

AIX

## Connexion de dispositifs de télémétrie à un gestionnaire de files d'attente

Les dispositifs de télémétrie se connectent à un gestionnaire de files d'attente à l'aide du client MQTT version 3. Le client MQTT version 3 utilise TCP/IP pour se connecter à un programme d'écoute TCP/IP appelé service de télémétrie (MQXR).

Lorsque vous connectez un dispositif de télémétrie à un gestionnaire de files d'attente, le client MQTT démarre une connexion TCP/IP en utilisant la méthode `MqttClient.connect`. A l'instar des clients IBM MQ, un client MQTT doit être connecté au gestionnaire de files d'attente pour envoyer et recevoir des messages. La connexion est effectuée au niveau du serveur à l'aide du programme d'écoute TCP/IP installé avec MQ Telemetry, appelé service de télémétrie (MQXR). Chaque gestionnaire de files d'attente exécute au maximum un service de télémétrie (MQXR).

Le service de télémétrie (MQXR) utilise l'adresse de connecteur à distance définie par chaque client dans la méthode `MqttClient.connect` pour allouer la connexion à un canal de télémétrie. Une adresse de connecteur est une combinaison de nom d'hôte TCP/IP et de numéro de port. Des clients multiples qui utilisent la même adresse de connecteur éloigné sont connectés au même canal de télémétrie par le service de télémétrie (MQXR).

S'il existe plusieurs gestionnaires de files d'attente sur un serveur, partagez les canaux de télémétrie entre les gestionnaires de files d'attente. Affectez les adresses de connecteur distant entre les gestionnaires de files d'attente. Définissez chaque canal de télémétrie à l'aide d'une adresse de connecteur unique. Deux canaux de télémétrie ne doivent pas utiliser la même adresse de connecteur.

Si la même adresse de connecteur à distance est configurée pour les canaux de télémétrie sur plusieurs gestionnaires de files d'attente, le premier canal de télémétrie à connecter l'emporte. Les canaux suivants qui se connectent à la même adresse échouent.

S'il existe plusieurs adaptateurs réseau sur le serveur, séparez les adresses de connecteur à distance entre les canaux de télémétrie. L'affectation des adresses de connecteur est totalement arbitraire, dans la mesure où toute adresse de connecteur particulière n'est configurée que sur un seul canal de télémétrie.

Configurez IBM MQ pour connecter les clients MQTT en utilisant les assistants fournis dans le supplément MQ Telemetry pour IBM MQ Explorer. Vous pouvez également suivre les instructions indiquées dans les sections [Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Linux et AIX](#) et [Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Windows](#) pour configurer manuellement la télémétrie.

### Référence associée

[Propriétés de MQXR](#)

Windows

Linux

AIX

## Protocoles de connexion pour la télémétrie

MQ Telemetry prend en charge IPv4 et IPv6, et TLS.

Windows

Linux

AIX

## Service de télémétrie (MQXR)

Le service de télémétrie (MQXR) est un programme d'écoute TCP/IP qui est géré comme un service IBM MQ. Créez le service à l'aide d'un assistant IBM MQ Explorer ou avec une commande `runmqsc`.

Le service MQ Telemetry (MQXR) est appelé `SYSTEM.MQXR.SERVICE`.

L'assistant Modèle de configuration de télémétrie fourni dans la fonction MQ Telemetry pour IBM MQ Explorer, crée le service de télémétrie et un exemple de canal de télémétrie. Voir [Vérification de l'installation de MQ Telemetry en utilisant IBM MQ Explorer](#).

Créez le modèle de configuration à partir de la ligne de commande ; voir [Vérification de l'installation de MQ Telemetry à l'aide de la ligne de commande](#).

Le service de télémétrie (MQXR) démarre et s'arrête automatiquement avec le gestionnaire de files d'attente. Contrôlez le service à l'aide du dossier de services figurant dans IBM MQ Explorer. Pour afficher le service, vous devez cliquer sur l'icône pour arrêter le IBM MQ Explorer filtrage des objets SYSTEM à partir de l'affichage.

Pour un exemple de création manuelle du service, voir

- [Linux](#) [AIX](#) [Création du SYSTEM.MQXR.SERVICE sur Linux.](#)
- [Windows](#) [Création du SYSTEM.MQXR.SERVICE sur Windows.](#)

**V 9.2.4** A partir de la IBM MQ 9.2.4 , la création du SYSTEM.MQXR.SERVICE sur Linux et la création du SYSTEM.MQXR.SERVICE sur Windows sont mises à jour pour spécifier la clé par défaut qui requiert des phrases passe pour que les canaux TLS MQTT soient chiffrés. Pour plus d'informations, voir [Chiffrement des phrases secrètes pour les canaux TLS MQTT](#).

## [Windows](#) [Linux](#) [AIX](#) **Canaux de télémétrie**

Créez des canaux de télémétrie pour créer des connexions avec des propriétés différentes, telles que l'authentification JAAS (Java Authentication and Authorization Service) ou TLS, ou pour gérer des groupes de clients.

Créez des canaux de télémétrie à l'aide de l'assistant **New Telemetry Channel** , fourni dans la fonction MQ Telemetry pour IBM MQ Explorer. Configurez un canal, en utilisant l'assistant, pour accepter les connexions depuis les clients MQTT sur un port TCP/IP donné. Depuis IBM WebSphere MQ 7.1, vous pouvez configurer MQ Telemetry en utilisant le programme de ligne de commande **runmqsc**.

Créez plusieurs canaux de télémétrie sur différents ports pour faciliter la gestion d'un grand nombre de connexions client, en divisant les clients en groupes. Chaque canal de télémétrie possède un nom différent.

Vous pouvez configurer les canaux de télémétrie avec des attributs de sécurité différents pour créer des types de connexion différents. Créez plusieurs canaux pour accepter des connexions client sur des adresses TCP/IP différentes. Utilisez TLS pour chiffrer les messages et authentifier le canal de télémétrie et le client. Voir [Configuration TLS des clients MQTT et des canaux de télémétrie](#). Spécifiez un ID utilisateur pour simplifier l'autorisation d'accès aux objets IBM MQ. Spécifiez une configuration JAAS pour authentifier l'utilisateur MQTT avec JAAS. Voir [MQTT client identification, authorization, and authentication](#).

## [Windows](#) [Linux](#) [AIX](#) **Protocole IBM MQ Telemetry Transport**

Le protocole IBM MQ Telemetry Transport (MQTT) v3 est conçu pour l'échange de messages entre des petits périphériques sur une faible bande passante, ou pour les connexions coûteuses et envoyer des messages de façon fiable. Il utilise TCP/IP.

Le MQTT protocol est publié ; voir [Format et protocole IBM MQ Telemetry Transport](#). La version 3 du protocole utilise la publication/abonnement et prend en charge trois qualités de service : *autonome après diffusion, au moins une fois et une seule fois*.

La taille réduite des en-têtes de protocole et la charge de message du tableau d'octets permet de conserver la taille réduite du message. Les en-têtes comprennent un en-tête à deux octets fixes et jusqu'à 12 octets d'en-têtes variables supplémentaires. Le protocole utilise les en-têtes variables à 12 octets pour l'abonnement et la connexion, et uniquement 2 en-têtes variables de 2 octets pour la plupart des publications.

Avec trois qualités de service, vous pouvez établir un compromis entre la faible latence et la fiabilité. Voir [Qualities of service provided by an MQTT client](#). La qualité de service *autonome après diffusion* n'utilise aucun stockage de dispositif persistant et ne prend en charge qu'une seule transmission pour envoyer ou recevoir une publication. Les qualités de service *au moins une fois et une seule fois* nécessitent le

stockage persistant sur l'unité pour conserver l'état du protocole et sauvegarder un message jusqu'à la réception de l'accusé de réception.

Windows

Linux

AIX

## Clients MQTT

Une application client MQTT permet la collecte d'informations provenant du dispositif de télémétrie, la connexion au serveur et la publication d'informations sur le serveur. L'application peut également s'abonner à des rubriques, recevoir des publications et contrôler le dispositif de télémétrie.

Contrairement aux applications client IBM MQ, les clients MQTT ne sont pas des applications IBM MQ. Ils n'indiquent pas un gestionnaire de files d'attente auquel se connecter. Ils ne sont pas limités à l'utilisation d'interfaces de programmation IBM MQ. Au contraire, les clients MQTT implémentent le protocole MQTT 3. Vous pouvez écrire votre propre bibliothèque client pour l'interface avec le MQTT protocol dans le langage de programmation et sur la plateforme de votre choix. Voir [Format et protocole IBM MQ Telemetry Transport](#).

Pour simplifier l'écriture d'applications client MQTT, utilisez les bibliothèques client C, Java et JavaScript qui encapsulent le MQTT protocol d'un certain nombre de plateformes. Si vous incorporez ces bibliothèques dans vos applications MQTT, un client MQTT complètement fonctionnel peut comporter seulement 15 lignes de code. Les bibliothèques client MQTT sont disponibles gratuitement sur Eclipse Paho et MQTT.org. Voir [IBM MQ Telemetry Transport sample programs](#).

L'application client MQTT est toujours responsable de l'initiation d'une connexion avec un canal de télémétrie. Une fois connectée, l'application client MQTT ou une application IBM MQ peut commencer à échanger des messages.

Les applications client MQTT et les applications IBM MQ diffusent des publications et s'abonnent au même ensemble de rubriques. Une application IBM MQ peut également envoyer un message directement à une application client MQTT sans que cette dernière ait besoin de créer préalablement un abonnement. Voir [Configurer la mise en file d'attente distribuée pour envoyer des messages aux clients MQTT](#).

Les applications client MQTT sont connectées à IBM MQ via un canal de télémétrie. Ce canal fait office de pont entre les différents types de message utilisés par MQTT et IBM MQ. Il crée des publications et des abonnements dans le gestionnaire de files d'attente pour le compte de l'application client MQTT. Le canal de télémétrie envoie des publications qui correspondent aux abonnements d'une application client MQTT d'un gestionnaire de files d'attente vers l'application client MQTT.

Windows

Linux

AIX

## Envoi d'un message à un client MQTT

Les applications IBM MQ peuvent envoyer des messages aux clients MQTT v3 en les publiant dans les abonnements créés par les clients, ou envoyer les messages directement. Les clients MQTT peuvent s'envoyer des messages en les publiant dans les rubriques auxquelles sont abonnés les autres clients.

### Un client MQTT s'abonne à une publication qu'il reçoit d'IBM MQ.

Réalisez la tâche, [«Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer»](#), à la page 130 pour envoyer une publication d'IBM MQ à un client MQTT.

Un client MQTT v3 reçoit des messages en créant généralement un abonnement à une rubrique ou groupe de rubriques. Dans l'exemple de fragment de code, [Figure 45](#), à la page 129, le client MQTT s'abonne en utilisant la chaîne de rubrique "MQTT Examples". Une application IBM MQ C, [Figure 46](#), à la page 130, publie dans la rubrique à l'aide de la chaîne de rubrique "MQTT Examples". Dans le fragment de code à la [Figure 47](#), à la page 130, le client MQTT reçoit la publication dans la méthode de rappel, `messageArrived`.

Pour plus d'informations sur la façon de configurer IBM MQ pour envoyer des publications en réponse aux abonnements des clients MQTT, voir [Publication d'un message en réponse à un abonnement client MQTT](#).

## Une application IBM MQ envoie un message directement à un client MQTT.

Réalisez la tâche «[Envoi d'un message à un client MQTT en utilisant IBM MQ Explorer](#)», à la page 135 pour envoyer un message directement d'IBM MQ à un client MQTT.

Un message envoyé de cette manière à un client MQTT s'appelle un message non sollicité. Les clients MQTT v3 reçoivent des messages non sollicités comme publications avec un nom de rubrique défini. Le service de télémétrie (MQXR) détermine le nom de la rubrique sur le nom de la file d'attente éloignée.

Pour plus d'informations sur la configuration d'IBM MQ pour envoyer des messages directement aux clients MQTT, voir [Envoi direct d'un message à un client](#).

## Un client MQTT publie un message.

Un client MQTT v3 peut publier un message reçu par un autre client MQTT v3, mais il ne peut pas envoyer de message non sollicité. Le fragment de code à la [Figure 48](#), à la page 130 montre comment un client MQTT v3 écrit en Java, publie un message.

En général, pour envoyer un message à un client MQTT v3, chaque client crée un abonnement à son propre identifiant `ClientIdentifier`. Réalisez la tâche «[Publication d'un message vers un client MQTT v3](#)», à la page 136 pour publier un message d'un client MQTT vers un autre client MQTT en utilisant `ClientIdentifier` comme chaîne de rubrique.

## Exemples de fragments de code

Le fragment de code à la [Figure 45](#), à la page 129 montre comment un client MQTT écrit en Java crée un abonnement. Il a également besoin d'une méthode de rappel, `messageArrived` pour recevoir les publications relatives à l'abonnement.

```
String    clientId = String.format("%-23.23s",
                                System.getProperty("user.name") + "_" +
                                (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient("localhost", clientId);
String topicString = "MQTT Examples";
int      QoS = 1;
client.subscribe(topicString, QoS);
```

*Figure 45. Abonné client MQTT v3*

Le fragment de code à la [Figure 46](#), à la page 130 montre comment une application IBM MQ écrite en C envoie une publication. Le fragment de code est extrait à partir de la tâche [Création d'un diffuseur de publications sur une rubrique variable](#).

```

/* Define and set variables to defaults */
/* Omitted lines declaring variables */
char * topicName = ""
char * topicString = "MQTT Examples"
char * publication = "Hello world!";
do {
    MQCONN(qMgrName, &Hconn, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    td.ObjectType = MQOT_TOPIC; /* Object is a topic */
    td.Version = MQOD_VERSION_4; /* Descriptor needs to be V4 */
    strncpy(td.ObjectName, topicName, MQ_TOPIC_NAME_LENGTH);
    td.ObjectString.VSPtr = topicString;
    td.ObjectString.VSLength = (MQLONG)strlen(topicString);
    MQOPEN(Hconn, &td, MQOO_OUTPUT | MQOO_FAIL_IF QUIESCING, &Hobj, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    pmo.Options = MQPMO_FAIL_IF QUIESCING | MQPMO_RETAIN;
    MQPUT(Hconn, &td, &md, &pmo, (MQLONG)strlen(publication)+1, publication, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQCLOSE(Hconn, &Hobj, MQCO_NONE, &CompCode, &Reason);
    if (CompCode != MQCC_OK) break;
    MQDISC(&Hconn, &CompCode, &Reason);
} while (0);

```

Figure 46. Abonné IBM MQ

Lorsque la publication arrive, le client MQTT appelle la méthode `messageArrived` de la classe `MqttCallback` du client d'application MQTT.

```

public class Callback implements MqttCallback {
    public void messageArrived(MqttTopic topic, MqttMessage message) {
        try {
            System.out.println("Message arrived: \"" + message.toString()
                + "\" on topic \"" + topic.toString() + "\"");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
// ... Other callback methods

```

Figure 47. `messageArrived` méthode

La [Figure 48](#), à la [page 130](#) montre MQTT v3 publiant un message vers l'abonnement créé dans la [Figure 45](#), à la [page 129](#).

```

String address = "localhost";
String clientId = String.format("%-23.23s",
    System.getProperty("user.name") + "-" +
    (UUID.randomUUID().toString()).trim()).replace('-', '_');
MqttClient client = new MqttClient(address, clientId);
String topicString = "MQTT Examples";
MqttTopic topic = client.getTopic(Example.topicString);
String publication = "Hello world";
MqttMessage message = new MqttMessage(publication.getBytes());
MqttDeliveryToken token = topic.publish(message);

```

Figure 48. Abonné client MQTT v3

## Windows Linux AIX Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer

Suivez les étapes de cette tâche pour publier un message en utilisant IBM MQ Explorer et vous abonner avec l'utilitaire client MQTT. Une tâche supplémentaire indique comment configurer un alias

de gestionnaire de files d'attente plutôt que de paramétrer la file d'attente de transmission sur SYSTEM.MQTT.TRANSMIT.QUEUE.

## Avant de commencer

La tâche suppose que vous connaissiez IBM MQ et IBM MQ Explorer et qu'IBM MQ et la fonction MQ Telemetry soient installés.

L'utilisateur qui crée les ressources du gestionnaire de files d'attente doit disposer de droits suffisants pour cette opération. À des fins de démonstration, l'ID utilisateur IBM MQ Explorer est supposé être membre du groupe mqm.

## Pourquoi et quand exécuter cette tâche

Dans cette tâche, vous créez une rubrique dans IBM MQ et vous vous y abonnez à l'aide de l'utilitaire client MQTT. Lorsque vous publiez dans la rubrique en utilisant IBM MQ Explorer, le client MQTT reçoit la publication.

## Procédure

Effectuez l'une des tâches suivantes :

- Vous avez installé MQ Telemetry, mais ne l'avez pas encore démarré. Effectuez la tâche suivante : [«Démarrage de la tâche sans service de télémétrie \(MQXR\) défini», à la page 131](#)
- Vous avez exécuté IBM MQ Telemetry auparavant, mais voulez utiliser un nouveau gestionnaire de files d'attente pour effectuer les démonstrations. Effectuez la tâche suivante : [«Démarrage de la tâche sans service de télémétrie \(MQXR\) défini», à la page 131](#)
- Vous souhaitez effectuer la tâche à l'aide d'un gestionnaire de files d'attente existant et dont les ressources de télémétrie ne sont pas définies. Vous ne souhaitez pas exécuter l'assistant **Définition d'un modèle de configuration**.
  - a. Effectuez l'une des tâches suivantes pour configurer la télémétrie :
    - [Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Linux et AIX](#)
    - [Configuration d'un gestionnaire de files d'attente pour la télémétrie sous Windows](#)
  - b. Effectuez la tâche suivante : [«Démarrage de la tâche avec un service de télémétrie \(MQXR\) en cours d'exécution», à la page 132](#)
- Si vous souhaitez effectuer la tâche à l'aide d'un gestionnaire de files d'attente existant dont les ressources de télémétrie sont déjà définies, suivez la procédure [«Démarrage de la tâche avec un service de télémétrie \(MQXR\) en cours d'exécution», à la page 132.](#)

## Que faire ensuite

Effectuez la procédure [«Envoi d'un message à un client MQTT en utilisant IBM MQ Explorer», à la page 135](#) pour envoyer un message directement à l'utilitaire client.

## **Démarrage de la tâche sans service de télémétrie (MQXR) défini**

Créez un gestionnaire de files d'attente et exécutez l'assistant **Définition d'un modèle de configuration** pour paramétrer les ressources de télémétrie exemples du gestionnaire de files d'attente. Publiez un message à l'aide d'IBM MQ Explorer et abonnez-vous à cette publication à l'aide de l'utilitaire client MQTT.

## Pourquoi et quand exécuter cette tâche

Lorsque vous définissez les ressources de télémétrie à l'aide de l'assistant **Définition d'un modèle de configuration**, ce dernier définit les droits de l'ID utilisateur invité. Envisagez soigneusement l'octroi des droits de cette manière à l'ID utilisateur invité. guest sous Windowset nobody sous Linuxsont autorisés à publier et à s'abonner à la racine de l'arborescence de rubriques et à placer des messages dans SYSTEM.MQTT.TRANSMIT.QUEUE.

L'assistant définit également la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE, ce qui peut interférer avec les applications exécutées sur un gestionnaire de files d'attente existant. Il est possible, mais laborieux, de configurer la télémétrie et de ne pas utiliser la file d'attente de transmission. Pour ce faire, effectuez la tâche «Utilisation d'un alias de gestionnaire de files d'attente», à la page 134. Cette tâche permet de créer un gestionnaire de files d'attente afin d'éviter toute interférence potentielle avec une file d'attente de transmission existante par défaut.

## Procédure

1. A l'aide d'IBM MQ Explorer, créez et lancez un nouveau gestionnaire de files d'attente.
  - a) Clic droit sur le dossier Queue Managers > **Nouveau** > **Gestionnaire de files d'attente...** Entrez un nom de gestionnaire de files d'attente, puis cliquez sur **Terminer**.  
Créez un nom de gestionnaire de files d'attente ; par exemple, MQTTQMGR.
2. Créez et lancez le service de télémétrie (MQXR), puis créez un modèle de canal de télémétrie.
  - a) Ouvrez le dossier Queue Managers\QmgrName\Telemetry.
  - b) Cliquez sur **Définir la configuration de l'exemple... > Terminer**  
Laissez la case à cocher **Lancer l'utilitaire client MQTT** sélectionnée.
3. Créez un abonnement pour MQTT Example à l'aide de l'utilitaire client MQTT .
  - a) Cliquez sur **Connect**.  
L' **historique client** enregistre un événement Connected .
  - b) Entrez MQTT Example dans la zone **Subscription \ Topic > Subscribe**.  
L' **historique client** enregistre un événement Subscribed .
4. Créez MQTTExampleTopic dans IBM MQ.
  - a) Clic droit sur le dossier Queue Managers\QmgrName\Topics dans **Explorateur MQ > Nouveau > Rubrique**.
  - b) Entrez MQTTExampleTopic dans la zone **Nom**, puis cliquez sur **Suivant**.
  - c) Entrez MQTT Example comme **Chaîne de rubrique > Terminer**.
  - d) Cliquez sur **OK** pour fermer la fenêtre d'accusé de réception.
5. Publiez Hello World! dans la rubrique MQTT Example à l'aide de IBM MQ Explorer.
  - a) Cliquez sur le dossier Queue Managers\QmgrName\Topics dans le fichier IBM MQ Explorer.
  - b) Clic droit sur MQTTExampleTopic > **Publication de test...**
  - c) Entrez Hello World! dans le champ **Données de message > Publier le message > Passer à la fenêtre MQTT Client Utility**.  
L' **historique client** enregistre un événement Received .

### **Démarrage de la tâche avec un service de télémétrie (MQXR) en cours d'exécution**

Créez un canal de télémétrie ainsi qu'une rubrique. Autorisez l'utilisateur à employer la rubrique et la file d'attente de transmission de télémétrie. Publiez un message à l'aide d'IBM MQ Explorer et abonnez-vous à cette publication à l'aide de l'utilitaire client MQTT.

## Avant de commencer

Dans cette version de la tâche, un gestionnaire de files d'attente, *QmgrName*, est défini et en cours d'exécution. Un service de télémétrie (MQXR) est défini et en cours d'exécution. Ce service de télémétrie (MQXR) peut avoir été créé manuellement ou à l'aide de l'assistant **Définition d'un modèle de configuration**.

## Pourquoi et quand exécuter cette tâche

Dans cette tâche, vous configurez un gestionnaire de files d'attente existant afin d'envoyer une publication à l'utilitaire client MQTT.

L'étape «1», à la page 133 de la tâche définit la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE, ce qui peut interférer avec les applications exécutées sur un gestionnaire de files d'attente existant. Il est possible, mais laborieux, de configurer la télémétrie et de ne pas utiliser la file d'attente de transmission. Pour ce faire, effectuez la tâche «Utilisation d'un alias de gestionnaire de files d'attente», à la page 134.

## Procédure

1. Définissez SYSTEM.MQTT.TRANSMIT.QUEUE comme file d'attente de transmission par défaut.
  - a) Clic droit sur `Queue Managers\QmgrName folder` > **Propriétés...**
  - b) Cliquez sur **Communication** dans le navigateur.
  - c) Cliquez sur **Sélectionner ...** > Sélectionner SYSTEM.MQTT.TRANSMIT.QUEUE > **OK** > **OK**.
2. Créez un canal de télémétrie MQTTExampleChannel pour connecter l'utilitaire client MQTT à IBM MQ, et démarrez l'utilitaire client MQTT.
  - a) Clic droit sur le dossier `Queue Managers\QmgrName \Telemetry\Channels` dans **Explorateur MQ** > **Nouveau** > **Canal de télémétrie...**
  - b) Entrez MQTTExampleChannel dans la zone **Nom du canal**, puis cliquez sur **Suivant**, et à nouveau sur **Suivant**.
  - c) Remplacez l'**ID utilisateur fixe** sur le panneau d'autorisation du client par l'ID utilisateur qui va publier et s'abonner à MQTTExample > **Suivant**.
  - d) Laissez l'option **Lancer l'utilitaire client** sélectionnée, puis cliquez sur **Terminer**.
3. Créez un abonnement pour MQTT Example à l'aide de l'utilitaire client MQTT.
  - a) Cliquez sur **Connect**.  
L' **historique client** enregistre un événement Connected .
  - b) Entrez MQTT Example dans la zone **Subscription \ Topic** > **Subscribe**.  
L' **historique client** enregistre un événement Subscribed .
4. Créez MQTTExampleTopic dans IBM MQ.
  - a) Clic droit sur le dossier `Queue Managers\QmgrName\Topics` dans **Explorateur MQ** > **Nouveau** > **Rubrique**.
  - b) Entrez MQTTExampleTopic dans la zone **Nom**, puis cliquez sur **Suivant**.
  - c) Entrez MQTT Example comme **Chaîne de rubrique** > **Terminer**.
  - d) Cliquez sur **OK** pour fermer la fenêtre d'accusé de réception.
5. Si vous souhaitez qu'un utilisateur, et non le groupe mqm, publie et s'abonne à la rubrique MQTTExample, procédez de la manière suivante :
  - a) Autorisez l'utilisateur à publier et à s'abonner à la rubrique MQTTExampleTopic :

```
setmqaut -m qMgrName -t topic -n MQTTExampleTopic -p User ID -all +pub +sub
```
  - b) Autorisez l'utilisateur à entrer un message sur SYSTEM.MQTT.TRANSMIT.QUEUE :

```
setmqaut -m qMgrName -t q -n SYSTEM.MQTT.TRANSMIT.QUEUE -p User ID -all +put
```
6. Publiez Hello World! dans la rubrique MQTT Example à l'aide de IBM MQ Explorer.
  - a) Cliquez sur le dossier `Queue Managers\QmgrName\Topics` dans le fichier IBM MQ Explorer.
  - b) Clic droit sur MQTTExampleTopic > **Publication de test...**

- c) Entrez Hello World! dans le champ **Données de message** > **Publier le message** > Passer à la fenêtre MQTT Client Utility.

L' **historique client** enregistre un événement Received .

### **Utilisation d'un alias de gestionnaire de files d'attente**

Publiez un message dans l'utilitaire client MQTT à l'aide de IBM MQ Explorer sans définir la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE.

Cette tâche est la suite de la précédente. Elle utilise un alias de gestionnaire de files d'attente afin d'éviter le paramétrage de la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE.

### **Avant de commencer**

Effectuez la tâche «Démarrage de la tâche sans service de télémétrie (MQXR) défini», à la page 131 ou «Démarrage de la tâche avec un service de télémétrie (MQXR) en cours d'exécution», à la page 132.

### **Pourquoi et quand exécuter cette tâche**

Lorsqu'un client MQTT crée un abonnement, IBM MQ envoie sa réponse en utilisant ClientIdentifieur comme nom de gestionnaire de files d'attente éloignées. Dans cette tâche, il utilise ClientIdentifieur, MyClient.

S'il n'y a pas de file d'attente de transmission ou d'alias de gestionnaire de files d'attente appelé MyClient, la réponse est placée dans la file d'attente de transmission par défaut. En définissant la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE, le client MQTT obtient la réponse.

Pour éviter de paramétrer la file d'attente de transmission sur SYSTEM.MQTT.TRANSMIT.QUEUE, utilisez les alias de gestionnaire de files d'attente. Vous devez configurer un alias de gestionnaire de files d'attente pour chaque ClientIdentifieur. En général, le nombre trop important de clients rend difficile l'utilisation d'alias de gestionnaire de files d'attente. ClientIdentifieur est souvent imprévisible, ce qui rend impossible la configuration de la télémétrie par cette méthode.

Il peut néanmoins arriver que vous ayez besoin de configurer la file d'attente de transmission par défaut sur une valeur autre que SYSTEM.MQTT.TRANSMIT.QUEUE. Les étapes de la [procédure](#) permettent de configurer un alias de gestionnaire de files d'attente au lieu de définir la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE.

### **Procédure**

1. Supprimez SYSTEM.MQTT.TRANSMIT.QUEUE comme file d'attente de transmission par défaut.
  - a) Clic droit sur Queue Managers\QmgrName folder > **Propriétés...**
  - b) Cliquez sur **Communication** dans le navigateur.
  - c) Retirez SYSTEM.MQTT.TRANSMIT.QUEUE de la zone **File d'attente de transmission par défaut**, puis cliquez sur **OK**.
2. Vérifiez que vous ne pouvez plus créer d'abonnement avec l'utilitaire client MQTT :
  - a) Cliquez sur **Connect**.

L' **historique client** enregistre un événement Connected .
  - b) Entrez MQTT Example dans la zone **Subscription \ Topic** > **Subscribe**.

L' **historique client** enregistre un événement Subscribe failed et un événement Connection lost .
3. Créez un alias de gestionnaire de files d'attente pour ClientIdentifieur, MyClient.
  - a) Cliquez avec le bouton droit de la souris sur le dossier Queue Managers\QmgrName\Queues > **Nouveau** > **Définition de file d'attente à distance**.
  - b) Nommez la définition MyClient, puis cliquez sur **Suivant**.

- c) Entrez MyClient dans la zone **Gestionnaire de files d'attente éloignées**.
  - d) Entrez SYSTEM.MQTT.TRANSMIT.QUEUE dans la zone **File d'attente de transmission**, puis cliquez sur **Terminer**.
4. Reconnectez l'utilitaire client MQTT.
- a) Vérifiez que l'option **Identificateur du client** est définie sur MyClient.
  - b) **Connecter**
- L' **historique client** enregistre un événement Connected .
5. Créez un abonnement pour MQTT Example à l'aide de l'utilitaire client MQTT .
- a) Cliquez sur **Connect**.
- L' **historique client** enregistre un événement Connected .
- b) Entrez MQTT Example dans la zone **Subscription \ Topic > Subscribe**.
- L' **historique client** enregistre un événement Subscribed .
6. Publiez Hello World! dans la rubrique MQTT Example à l'aide de IBM MQ Explorer.
- a) Cliquez sur le dossier Queue Managers\QmgrName\Topics dans le fichier IBM MQ Explorer.
  - b) Clic droit sur MQTTExampleTopic > **Publication de test...**
  - c) Entrez Hello World! dans le champ **Données de message > Publier le message > Passer à la fenêtre MQTT Client Utility**.
- L' **historique client** enregistre un événement Received .

## Windows Linux AIX Envoi d'un message à un client MQTT en utilisant IBM MQ Explorer

Envoyez un message à l'utilitaire client MQTT en plaçant un message dans la file d'attente IBM MQ avec IBM MQ Explorer. Cette tâche indique comment configurer une définition de file d'attente éloignée pour envoyer un message directement à un client MQTT.

### Avant de commencer

Exécutez la tâche «[Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer](#)», à la page 130. Maintenez l'utilitaire client MQTT connecté.

### Pourquoi et quand exécuter cette tâche

Cette tâche illustre l'envoi d'un message à un client MQTT à l'aide d'une file d'attente plutôt que par la publication dans une rubrique. Vous ne créez pas d'abonnement dans le client. L'étape «2», à la page 136 de la tâche indique que l'abonnement précédent a été supprimé.

### Procédure

1. Supprimez les abonnements existants en déconnectant, puis en reconnectant l'utilitaire client MQTT.
 

L'abonnement est supprimé, car (à moins que vous ne changiez les valeurs par défaut) l'utilitaire client MQTT se connecte avec une session propre. Voir [Sessions propres](#).

Pour faciliter cette tâche, entrez votre propre identifiant ClientIdentifieur au lieu d'utiliser l'identifiant ClientIdentifieur créé par l'utilitaire client MQTT.

  - a) Cliquez sur l'option de **déconnexion** pour déconnecter l'utilitaire client MQTT du canal de télémétrie.

L' **historique client** enregistre un événement Disconnected

  - b) Remplacez l'identifiant **Identificateur de client** par MyClient.
  - c) Cliquez sur **Connect**.

- L' **historique client** enregistre un événement Connected
2. Vérifiez que l'utilitaire client MQTT ne reçoit plus de publication pour MQTTExampleTopic.
    - a) Cliquez sur le dossier Queue Managers\*QmgrName*\Topics dans le fichier IBM MQ Explorer.
    - b) Clic droit sur MQTTExampleTopic > **Publication de test...**
    - c) Entrez Hello World! dans le champ **Données de message** > **Publier le message** > Passer à la fenêtre MQTT Client Utility.

Aucun événement n'est enregistré dans l'**historique client**.
  3. Créez une définition de file d'attente éloignée pour le client.
 

Paramétrez l'identifiant ClientIdentifier, MyClient, en tant que nom de gestionnaire de files d'attente éloignées dans la définition de file d'attente éloignée. Utilisez le nom de votre choix en tant que nom de file d'attente éloignée. Ce nom est transmis à un client MQTT en tant que nom de rubrique.

    - a) Cliquez avec le bouton droit de la souris sur le dossier Queue Managers\*QmgrName*\Queues > **Nouveau** > **Définition de file d'attente à distance**.
    - b) Nommez la définition MyClientRemoteQueue, puis cliquez sur **Suivant**.
    - c) Entrez MQTTExampleQueue dans la zone **File d'attente éloignée**.
    - d) Entrez MyClient dans la zone **Gestionnaire de files d'attente éloignées**.
    - e) Entrez SYSTEM.MQTT.TRANSMIT.QUEUE dans la zone **File d'attente de transmission**, puis cliquez sur **Terminer**.
  4. Placez un message de test sur MyClientRemoteQueue.
    - a) Cliquez avec le bouton droit de la souris sur **MyClientRemoteQueue** > **Insérer un message de test...**
    - b) Entrez Hello queue! dans la zone de données Message > **Insérer un message** > **Fermer**

L' **historique client** enregistre un événement Received .
  5. Supprimez SYSTEM.MQTT.TRANSMIT.QUEUE comme file d'attente de transmission par défaut.
    - a) Clic droit sur Queue Managers\*QmgrName* folder > **Propriétés...**
    - b) Cliquez sur **Communication** dans le navigateur.
    - c) Retirez SYSTEM.MQTT.TRANSMIT.QUEUE de la zone **File d'attente de transmission par défaut**, puis cliquez sur **OK**.
  6. Réexécutez l'étape «4», à la page 136.
- MyClientRemoteQueue est une définition de file d'attente à distance qui désigne explicitement la file d'attente de transmission. Vous n'avez pas besoin de définir la file d'attente de transmission par défaut pour envoyer un message à MyClient.

## Que faire ensuite

Lorsque la file d'attente de transmission par défaut n'est plus définie sur SYSTEM.MQTT.TRANSMIT.QUEUE, l'utilitaire client MQTT ne peut pas créer d'abonnement sauf si un alias de gestionnaire de files d'attente est défini pour ClientIdentifier, MyClient. Restaurez la définition de la file d'attente de transmission par défaut sur SYSTEM.MQTT.TRANSMIT.QUEUE.

## Windows Linux AIX Publication d'un message vers un client MQTT v3

Publication d'un message d'un client MQTT v3 vers un autre en utilisant ClientIdentifier comme nom de rubrique et IBM MQ comme courtier de publication/abonnement.

### Avant de commencer

Exécutez la tâche «Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer», à la page 130. Maintenez l'utilitaire client MQTT connecté.

## Pourquoi et quand exécuter cette tâche

Cette tâche décrit deux opérations :

1. Abonnement à une rubrique dans un client MQTT et réception d'une publication d'un autre client MQTT.
2. Définition d'abonnements "point-à-point" en utilisant l'identificateur de client en tant que chaîne de rubrique.

## Procédure

1. Supprimez les abonnements existants en déconnectant, puis en reconnectant l'utilitaire client MQTT.  
L'abonnement est supprimé, car (à moins que vous ne changiez les valeurs par défaut) l'utilitaire client MQTT se connecte avec une session propre. Voir [Sessions propres](#).  
Pour faciliter cette tâche, entrez votre propre identifiant `ClientIdentifieur` au lieu d'utiliser l'identifiant `ClientIdentifieur` créé par l'utilitaire client MQTT.
  - a) Cliquez sur l'option de **déconnexion** pour déconnecter l'utilitaire client MQTT du canal de télémétrie.  
L' **historique client** enregistre un événement `Disconnected`
  - b) Remplacez l'identifiant **Identificateur de client** par `MyClient`.
  - c) Cliquez sur **Connect**.  
L' **historique client** enregistre un événement `Connected`
2. Créez un abonnement à la rubrique `MyClient`.  
`MyClient` est l'identificateur de client de ce client.
  - a) Entrez `MyClient` dans la zone **Abonnement\Rubrique**, puis cliquez sur **S'abonner**.  
L' **historique client** enregistre un événement `Subscribed`.
3. Démarrez un autre utilitaire client MQTT.
  - a) Ouvrez le dossier `Queue Managers\QmgrName\Telemetry\channels`.
  - b) Cliquez avec le bouton droit de la souris sur le canal **PlainText**, puis sélectionnez **Exécuter l'utilitaire client MQTT...**
  - c) Cliquez sur **Connect**.  
L' **historique client** enregistre un événement `Connected`
4. Publiez `Hello MyClient!` dans la rubrique `MyClient`.
  - a) Copiez la rubrique d'abonnement, `MyClient`, à partir de l'utilitaire client MQTT exécuté avec le `ClientIdentifieur`, `MyClient`.
  - b) Collez `MyClient` dans le champ **Publication\Topic** de chaque instance de l'utilitaire client MQTT.
  - c) Entrez `Hello MyClient!` dans la zone **Publication \ message**.
  - d) Cliquez sur **Publication** dans les deux instances.

## Résultats

L'**historique du client** dans l'utilitaire client MQTT avec le `ClientIdentifieur`, `MyClient`, enregistre deux événements **reçus** et un événement **publié**. L'autre instance d'utilitaire client MQTT enregistre un événement **Publié**.

Si un seul événement **Reçu** est affiché, les raisons peuvent en être les suivantes :

1. La file d'attente de transmission par défaut du gestionnaire de files d'attente est-elle définie sur `SYSTEM.MQTT.TRANSMIT.QUEUE` ?
2. Avez-vous créé des alias de gestionnaire de files d'attente ou des définitions de files d'attente à distance faisant référence à `MyClient` pour les autres exercices ? Si vous avez un problème de

configuration, supprimez toutes les ressources qui font référence à MyClient, telles qu'un alias de gestionnaire de files d'attente ou des files d'attente de transmission. Déconnectez les utilitaires client, arrêtez et redémarrez le service de télémétrie (MQXR).

## Windows Linux AIX **Envoi d'un message à une application IBM MQ depuis un client MQTT**

Une application IBM MQ peut recevoir un message depuis un client MQTT version 3 en s'abonnant à une rubrique. Le client MQTT se connecte IBM MQ en utilisant un canal de télémétrie et envoie un message à l'application IBM MQ en publiant dans la même rubrique.

Réalisez la tâche «[Publication d'un message vers IBM MQ depuis un client MQTT](#)», à la page 138 pour apprendre à envoyer une publication d'un client MQTT à un abonnement défini IBM MQ.

Si la rubrique est mise en cluster ou répartie à l'aide d'une hiérarchie publication/abonnement, l'abonnement peut se trouver dans un gestionnaire de files d'attente différent de celui auquel le client MQTT est connecté.

## Windows Linux AIX **Publication d'un message vers IBM MQ depuis un client MQTT**

Créez un abonnement à une rubrique à l'aide d'IBM MQ Explorer et publiez des données dans la rubrique à l'aide de l'utilitaire client MQTT.

### **Avant de commencer**

Exécutez la tâche «[Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer](#)», à la page 130. Maintenez l'utilitaire client MQTT connecté.

### **Pourquoi et quand exécuter cette tâche**

La tâche illustre la publication d'un message avec un client MQTT et la réception de la publication à l'aide d'un abonnement durable non géré créé en utilisant IBM MQ Explorer.

### **Procédure**

1. Créez un abonnement durable à la chaîne de rubrique MQTT Exemple.

Exécutez les étapes suivantes pour créer la file d'attente et l'abonnement en utilisant IBM MQ Explorer.

- a) Cliquez avec le bouton droit de la souris sur le dossier Queue Managers\*QmgrName*\Queues dans IBM MQ Explorer > **Nouveau** > **File d'attente locale...**
- b) Entrez MQTTExampleQueue comme nom de la file d'attente, puis cliquez sur **Terminer**.
- c) Cliquez avec le bouton droit de la souris sur le dossier Queue Managers\*QmgrName*\Subscriptions dans IBM MQ Explorer > **Nouveau** > **Abonnement...**
- d) Entrez MQTTExampleSubscription comme nom de la file d'attente, puis cliquez sur **Suivant**.
- e) Cliquez sur **Sélectionner...** > MQTTExampleTopic > **OK**.

Vous avez déjà créé la rubrique MQTTExampleTopic à l'étape «4», à la page 132 de «[Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer](#)», à la page 130.

- f) Entrez MQTTExampleQueue comme nom de destination, puis cliquez sur **Terminer**.
2. En tant qu'étape facultative, définissez la file d'attente à utiliser par un autre utilisateur, sans autorisation mqm.

Si vous paramétrez la configuration pour les utilisateurs disposant de moins de droits que mqm, vous devez attribuer les droits put et get à MQTTExampleQueue. L'accès à la rubrique et à la file d'attente de transmission a été configuré à la section «[Publication d'un message dans l'utilitaire client MQTT depuis IBM MQ Explorer](#)», à la page 130.

- a) Autoriser un utilisateur à placer des éléments dans la file d'attente MQTTExampleQueue et à y accéder :

```
setmqaut -m qMgrName -t queue -n MQTTExampleQueue -p User ID -all +put +get
```

3. Publiez Hello IBM MQ! dans la rubrique MQTT Example à l'aide de l'utilitaire client MQTT .

Si vous n'avez pas quitté l'utilitaire client MQTT connecté, cliquez avec le bouton droit de la souris sur le canal **Texte plan > Exécuter MQTT Utilitaire client... > Connecter**.

- a) Entrez MQTT Example dans la zone **Publication\Rubrique**.  
b) Entrez Hello IBM MQ! dans le champ **Publication\Message > Publier**.
4. Ouvrez le dossier Queue Managers\QmgrName\Queues et recherchez MQTTExampleQueue.

La valeur entrée dans la zone **Longueur de la file d'attente en cours** est 1.

5. Cliquez avec le bouton droit de la souris sur MQTTExampleQueue > **Parcourir les messages ...** et examinez la publication.

Windows

Linux

AIX

## Applications de publication/abonnement MQTT

Utilisez la publication/l'abonnement basée sur une rubrique pour écrire des applications MQTT.

Lorsque le client MQTT est connecté, les publications transitent dans l'une ou l'autre direction entre le client et le serveur. Les publications sont envoyées à partir du client où les informations sont publiées côté client. Les publications sont reçues du côté du client lorsque le message est publié sur une rubrique qui correspond à une description créée par le client.

Le courtier de publication/abonnement IBM MQ gère les rubriques et les abonnements créés par les clients MQTT. Les rubriques créées par les clients MQTT partagent le même espace de sujet que les rubriques créées par les applications IBM MQ.

Les publications qui correspondent à la chaîne de rubrique dans un abonnement de client MQTT sont placées sur SYSTEM.MQTT.TRANSMIT.QUEUE avec le nom du gestionnaire de files d'attente éloignées défini sur ClientIdentifieur du client. Le service de télémétrie (MQXR) transfère les publications au client qui a créé l'abonnement. Pour identifier le client, il utilise l'identificateur de client défini en tant que nom du gestionnaire de files d'attentes éloignées.

En règle générale, SYSTEM.MQTT.TRANSMIT.QUEUE doit être défini en tant que file d'attente de transmission par défaut. Il est possible, mais onéreux, de configurer MQTT pour ne pas utiliser la file d'attente de transmission par défaut. Voir [Configuration des files d'attente réparties pour envoyer des messages aux clients MQTT](#).

Un client MQTT peut créer une session persistante. Voir «Sessions avec état et sans état MQTT», à la page 143. Les abonnements créés dans une session persistante sont durables. Les publications qui arrivent pour un client avec une session persistante sont stockées dans SYSTEM.MQTT.TRANSMIT.QUEUE, et transmises au client lors de la reconnexion.

Un client MQTT peut également publier et s'abonner à des publications conservées. Voir [Publications conservées et clients MQTT](#). Un abonné à une publication conservée reçoit la dernière publication dans la rubrique. Il la reçoit lorsqu'il crée un abonnement ou lorsqu'il se reconnecte à la session précédente.

Windows

Linux

AIX

## Applications de télémétrie

Ecrivez des applications de télémétrie à l'aide de flux de messages IBM MQ ou IBM Integration Bus.

Utilisez JMS, MQI ou d'autres interfaces de programmation IBM MQ pour programmer des applications de télémétrie dans IBM MQ.

Le service de télémétrie (MQXR) convertit entre les messages MQTT v3 et les messages IBM MQ. Il crée des abonnements et des publications pour les clients MQTT et envoie les publications aux clients MQTT. Une publication est la charge d'un message MQTT v3. La charge comprend les en-têtes de message et un tableau d'octets au format jms-bytes. Le serveur de télémétrie mappe les en-têtes entre un message

MQTT v3 et un message IBM MQ. Voir [«Intégration de MQ Telemetry aux gestionnaires de files d'attente»](#), à la page 140.

Utilisez les noeuds Publication, MQInput et JMSInput pour envoyer et recevoir des publications entre IBM Integration Bus et les clients MQTT.

L'utilisation de flux de messages permet d'intégrer la télémétrie aux sites Web à l'aide de HTTP et à d'autres applications en utilisant des adaptateurs IBM MQ et WebSphere.

## Windows Linux AIX Intégration de MQ Telemetry aux gestionnaires de files d'attente

Le client MQTT est intégré à IBM MQ comme application de publication/abonnement. Il peut publier ou s'abonner à des rubriques dans IBM MQ, en créant des rubriques, ou en utilisant des rubriques existantes. Il reçoit les publications d'IBM MQ comme résultat des clients MQTT, y compris de lui-même, ou des autres applications IBM MQ qui publient dans les rubriques de ses abonnements. Les règles sont appliquées pour décider des attributs d'une publication.

La plupart des attributs associés aux rubriques, publications, abonnements et messages fournis par IBM MQ ne sont pas pris en charge. Les sections [«Client MQTT vers un courtier de publication/abonnement IBM MQ»](#), à la page 140 et [«IBM MQ vers un client MQTT»](#), à la page 142 décrivent comment sont définis les attributs des publications. Les paramètres varient selon que la publication est envoyée ou provient d'un courtier de publication/abonnement IBM MQ.

Dans IBM MQ, les rubriques de publication/abonnement sont associées à des objets de rubrique d'administration. Les rubriques créées par les clients MQTT ne sont pas différentes. Lorsqu'un client MQTT crée une chaîne de rubrique pour une publication, le courtier de publication/abonnement IBM MQ l'associe à un objet de rubrique d'administration. Le courtier mappe la chaîne de rubrique dans la publication avec le plus proche parent d'objet de rubrique d'administration. Le mappage est le même que celui utilisé pour les applications IBM MQ. S'il n'existe aucune rubrique créée par l'utilisateur, la publication est mappée à SYSTEM.BASE.TOPIC. Les attributs qui sont appliqués à la publication sont dérivés de l'objet de rubrique.

Lorsqu'une application IBM MQ ou un administrateur crée un abonnement, l'abonnement est nommé. Répertorie les abonnements à l'aide de IBM MQ Explorer, ou à l'aide de commandes `runmqsc` ou PCF. Tous les abonnements client MQTT sont nommés. Ils reçoivent un nom au format suivant:  
*ClientIdentifier:Topic name*

### Client MQTT vers un courtier de publication/abonnement IBM MQ

Un client MQTT a envoyé une publication à IBM MQ. Le service de télémétrie (MQXR) convertit la publication en message IBM MQ. Le message IBM MQ contient trois parties :

1. MQMD
2. RFH2
3. Message

Les propriétés de MQMD sont définies à leurs valeurs par défaut, sauf lorsqu'elles sont inscrites dans le Tableau 9, à la page 140.

Zone MQMD	Tapez	Valeur
Format	MQCHAR8	MQFMT_RF_HEADER_2

Tableau 9. Paramètres pour les propriétés MQMD (suite)

Zone MQMD	Tapez	Valeur
<b>UserIdentif</b>	MQCHAR12	Prend l'une des valeurs : MqttClient.ClientIdentif MqttConnectOptions.UserName ID utilisateur défini par l'administrateur IBM MQ pour le canal de télémétrie.
<b>Priority</b>	MQLONG	MQPRI_PRIORITY_AS_Q_DEF (valeur par défaut d'IBM MQ, qui est différente de celle de JMS, qui est 4.)
<b>Persistence</b>	MQLONG	QoS=0→MQPER_NOT_PERSISTENT QoS=1→MQPER_PERSISTENT QoS=2→MQPER_PERSISTENT

L'en-tête RFH2 ne contient pas de dossier <msd> définissant le type du message JMS. Le service de télémétrie crée le message IBM MQ comme message par défaut JMS. Le type de message par défaut JMS est un message jms-bytes. Une application peut accéder à d'autres informations d'en-tête en tant que propriétés de message ; voir [Propriétés des messages](#).

Les valeurs RFH2 sont définies comme indiqué dans le [Tableau 10](#), à la page 141. La propriété Format est définie dans l'en-tête fixe RFH2 et les autres valeurs sont définies dans les dossiers RFH2.

Tableau 10. Paramètres pour les propriétés RFH2

Propriété RFH2	Type/Dossier	En-tête
Format	MQCHAR8	MQFMT_NONE
ClientIdentif	mqtt/clientId	Copiez MqttClient.ClientIdentif avec une longueur de 1...23 octets.
QoS	mqtt/qos	Copiez QoS depuis le message entrant MQTT.
ID message	mqtt/msgid	Copiez l'ID de message du message entrant MQTT si QoS a la valeur 1 ou 2.
MQIsRetained	mmps/Ret	Définissez cette propriété si la publication MQTT d'origine a été envoyée avec la propriété RETAIN définie et que le message est reçu comme publication conservée.
MQTopicString	mmps/Top	Rubrique dans laquelle le message MQTT a été publié.

La charge dans une publication MQTT est mappée au contenu du message IBM MQ :

Tableau 11. Mappage du contenu d'une publication MQTT au contenu d'un message IBM MQ

Contenu du message	Tapez	Contenu du message IBM MQ
Tampon	MQBYTE <i>n</i>	Copiez les octets du message entrant MQTT. La longueur peut être zéro.

## IBM MQ vers un client MQTT

Un client s'est abonné à une rubrique de publication. Une application IBM MQ a publié dans la rubrique, et une publication a été envoyée à l'abonné MQTT par le courtier de publication/abonnement IBM MQ. Ou bien, une application IBM MQ a envoyé un message non sollicité directement à un client MQTT. Le Tableau 12, à la page 142 explique comment les en-têtes de message fixes sont définis dans le message envoyé au client MQTT. Les autres données dans l'en-tête de message IBM MQ ou les autres en-têtes sont supprimés. Les données dans le message IBM MQ sont envoyées comme charge de message dans le message MQTT sans modification. Le message MQTT est envoyé au client MQTT par le service de télémétrie (MQXR).

*Tableau 12. Définition des en-têtes de message fixes dans un message IBM MQ envoyé au client MQTT*

MQTTZone	Tapez	Valeur
<b>DUP</b>	Booléen	Défini si QoS = 1 ou 2, et que le message a été envoyé à ce client lors d'une transmission précédente, et que ce message n'a pas reçu d'accusé de réception après un certain laps de temps.
<b>QoS</b>	entier	<p>La manière dont la valeur de QoS dans la publication sortante du courtier de publication/abonnement IBM MQ est définie dépend de la publication entrante. Elle varie selon que la publication entrante a été envoyée par un client MQTT ou une application IBM MQ.</p> <p><b>MQTT</b> Valeur la plus faible de qualité de service QoS dans la publication entrante et dans la QoS demandé par l'abonné.</p> <p><b>IBM MQ</b> Valeur la plus faible de QoS dérivée de la publication entrante :</p> <p style="padding-left: 40px;">MQPER_NOT_PERSISTENT→QoS=0 MQPER_PERSISTENT→QoS=2</p> <p>et la QoS demandée par l'abonné. Si le message est envoyé au client sans un abonnement, QoS prend la valeur par défaut 2. Un client peut modifier cette valeur en s'abonnant à DEFAULT . QoS avec une QoS différente.</p>
<b>RETAIN</b>	Booléen	Défini si la propriété retained a été définie pour la publication.

Le Tableau 13, à la page 142 explique comment les en-têtes de message variables sont définis dans le message MQTT envoyé au client MQTT.

*Tableau 13. Définition des propriétés d'en-tête variable MQTT dans un message MQTT envoyé au client MQTT*

MQTTZone	Tapez	Valeur
<b>Topic name</b>	Chaîne	Chaîne de rubrique avec laquelle le message a été publié.
<b>Message ID</b>	Chaîne	Deux derniers octets de la propriété MQMD . MsgId de la publication lorsqu'elle est placée dans SYSTEM . MQTT . TRANSMIT . QUEUE.
<b>Payload</b>	octet[]	Copie directe des octets de la publication entrante dans le courtier de publication/abonnement. La longueur peut être zéro.

## Sessions avec état et sans état MQTT

Les clients MQTT peuvent créer une session avec état avec le gestionnaire de files d'attente. Lorsqu'un client MQTT avec état se déconnecte, le gestionnaire de files d'attente conserve les abonnements créés par le client ainsi que les messages en cours. Lorsque le client se reconnecte, il le message en cours. Il envoie les messages mis en file d'attente pour être distribués et reçoit les messages publiés pour ses abonnements pendant qu'il était déconnecté.

Lorsqu'un client MQTT se connecte à un canal de télémétrie, il démarre une nouvelle session ou relance une session. Dans le cas d'une nouvelle session, il n'existe aucun message en attente n'ayant pas reçu d'accusé de réception, aucun abonnement et aucune publication en attente de distribution. Lorsqu'un client se connecte, il indique si une session propre doit être démarrée ou s'il convient de relancer une session existante ; voir [Sessions propres](#).

Si le client relance une session existante, celle-ci se poursuit comme si la connexion n'avait pas été interrompue. Les publications en attente de distribution sont envoyées au client et les réacheminements de messages qui n'avaient pas été validés sont terminés. Lorsqu'un client dans une session persistante se déconnecte du service de télémétrie (MQXR), les abonnements créés par le client sont conservés. Les publications des abonnements sont envoyées au client lorsqu'il se reconnecte. Si ce dernier se reconnecte sans relancer la précédente session, les publications sont supprimées par le service de télémétrie (MQXR).

Les informations d'état de session sont sauvegardées dans le gestionnaire de files d'attente SYSTEM.MQTT.PERSISTENT.STATE.

L'administrateur IBM MQ peut déconnecter et purger la session.

## Lorsqu'un client MQTT n'est pas connecté

Lorsqu'un client MQTT n'est pas connecté, le gestionnaire de files d'attente peut continuer à recevoir des publications en son nom. Elles sont réacheminées au client lorsque ce dernier se reconnecte. Un client peut créer une publication "Last will and testament" (Dernière volonté et testament) qui sera publiée par le gestionnaire de files d'attente au nom du client si ce dernier se déconnecte de manière inattendue.

Si vous voulez être prévenu lorsque le client se déconnecte de façon inattendue, vous pouvez enregistrer une publication "Last will and testament". Voir [Publication Last will and testament \(Dernière volonté et testament\)](#). Cette publication est envoyée au service de télémétrie (MQXR) s'il détecte que la connexion avec le client a été interrompue sans que le client en ait fait la demande.

Un client peut publier une publication conservée à tout moment ; voir [Publications conservées et clients MQTT](#). Un nouvel abonnement à une rubrique peut nécessiter l'envoi d'une publication conservée associée à cette publication. Si vous créez la publication "last will and testament" en tant que publication conservée, vous pouvez l'utiliser pour contrôler le statut du client.

Par exemple, le client diffuse une publication conservée, lorsqu'il se connecte, indiquant qu'elle est disponible. Au même moment, il crée une publication conservée "last will and testament" qui annonce sa disponibilité. De plus, juste avant d'effectuer une déconnexion planifiée, il publie sa non disponibilité en tant que publication conservée. Pour trouver si le client est disponible, vous devez vous abonner à la rubrique de la publication conservée. Vous devez toujours recevoir l'une des trois publications.

Si le client doit recevoir des messages publiés lorsqu'il est déconnecté, reconnectez le client à la session précédente ; voir «[Sessions avec état et sans état MQTT](#)», à la page 143. Ses abonnements sont actifs jusqu'à leur suppression ou jusqu'à ce que le client crée une session propre.

## Couplage non contraignant entre les clients MQTT et les applications IBM MQn

Le flux de publications entre les clients MQTT et les applications IBM MQ est couplé de façon non contraignante. Les publications peuvent provenir d'un client MQTT ou d'une application IBM MQ et dans un ordre non défini. Les diffuseurs de publications et les abonnés sont couplés de façon non

contraignante. Ils interagissent de manière indirecte via les publications et les abonnements. Vous pouvez aussi envoyer directement des messages à un client MQTT depuis une application IBM MQ.

Les clients MQTT et les applications IBM MQ sont couplés de façon non contraignante dans deux sens :

1. Les diffuseurs de publication et les abonnés sont couplés de façon non contraignantes via l'association d'une publication et d'un abonnement avec une rubrique. Les diffuseurs de publication et les abonnés ne connaissent généralement pas l'adresse ou l'identité de l'autre source de publication ou d'abonnement.
2. Les clients MQTT publient, reçoivent et s'abonnent à des publications, et ils traitent les accusés de réception de distribution sur des unités d'exécution distincte.

Une application client MQTT n'attend pas la distribution d'une publication. Elle transmet un message au client MQTT, puis l'application continue dans sa propre unité d'exécution. Un jeton de distribution est utilisé pour synchroniser l'application avec la distribution d'une publication ; voir [Jetons de distribution](#).

Une fois que le message a été transmis au client MQTT, l'application a le choix d'attendre le jeton de distribution. Au lieu d'attendre, le client peut fournir une méthode de rappel qui est appelée lorsque la publication est distribuée à IBM MQ. L'application peut également ignorer le jeton de distribution.

En fonction de la qualité de service associée au message, le jeton de distribution est renvoyé immédiatement à la méthode de rappel ou probablement après un délai très important. Le jeton de distribution peut même être renvoyé après déconnexion et reconnexion du client. Si la qualité de service est de type *autonome après diffusion*, le jeton de distribution est renvoyé immédiatement. Dans les deux autres cas, le jeton de distribution est renvoyé uniquement lorsque le client reçoit un accusé de réception indiquant que la publication a été envoyée aux abonnés.

Les publications envoyées au client MQTT à la suite d'un abonnement du client sont distribuées à la méthode de rappel `messageArrived`. `messageArrived` s'exécute sur une unité d'exécution différente de l'application principale.

## Envoi de messages directement à un client MQTT

Vous pouvez envoyer un message à un client MQTT de l'une des deux façons suivantes.

1. Une application IBM MQ peut envoyer un message directement à un client MQTT sans abonnement. Voir [Envoi direct d'un message à un client](#).
2. Vous pouvez également utiliser la convention de dénomination de votre `identificateur de client`. Demandez aux abonnés MQTT de créer des abonnements en utilisant leur identifiant `ClientIdentifier` unique comme rubrique. Diffusez la publication pour l'*identificateur de client*. La publication est envoyée au client qui s'est abonné à la rubrique *Identificateur de client*. Cette méthode permet d'envoyer une publication à un abonné MQTT.

Windows

Linux

AIX

## Sécurité MQ Telemetry

La sécurisation des dispositifs de télémétrie peut être cruciale étant donné que les dispositifs sont susceptibles d'être portables et utilisés dans des endroits qui ne sont pas correctement contrôlés. Vous pouvez utiliser un réseau privé virtuel (VPN) pour sécuriser la connexion du périphérique MQTT vers le service de télémétrie (MQXR). MQ Telemetry fournit deux autres mécanismes de sécurité, TLS et JAAS.

TLS est principalement utilisé pour chiffrer les communications entre le dispositif et le canal de télémétrie, et pour s'assurer que le dispositif est connecté au serveur approprié ; voir [Authentification du canal de télémétrie à l'aide de TLS](#). Vous pouvez également utiliser TLS pour vérifier que le dispositif client peut se connecter au serveur. Voir [Authentification du client MQTT](#).

JAAS est principalement utilisé pour vérifier que l'utilisateur du dispositif peut utiliser une application serveur. Voir [Authentification du client MQTT en utilisant un mot de passe](#). JAAS peut être utilisé avec LDAP pour vérifier un mot de passe à l'aide d'un répertoire de connexion unique.

TLS et JAAS peuvent être utilisés conjointement pour offrir deux facteurs d'authentification. Vous pouvez restreindre les chiffrements utilisés par TLS aux chiffrements respectant les normes FIPS.

Avec au moins dix ou des milliers d'utilisateurs, il n'est pas toujours pratique d'accorder des profils de sécurité individuels. Il n'est pas non plus pratique d'utiliser des profils pour autoriser les utilisateurs individuels à accéder aux objets IBM MQ. Regroupez plutôt les utilisateurs dans des classes pour l'autorisation de publication et d'abonnement aux rubriques et l'envoi de publications aux clients.

Configurez chaque canal de télémétrie pour mapper les clients à des ID utilisateur client communs. Servez-vous d'un ID utilisateur commun pour tout client qui se connecte sur un canal spécifique. Voir [Identité et autorisation du client MQTT](#).

L'autorisation de groupes d'utilisateurs n'a pas d'incidence sur l'authentification de chaque individu. Chaque utilisateur individuel peut être authentifié, côté client ou serveur avec son Nom d'utilisateur et son Mot de passe, puis être authentifié sur le serveur à l'aide d'un ID utilisateur commun.

Windows

Linux

AIX

## Globalisation MQ Telemetry

La charge de message dans le protocole MQTT v3 est codée comme un tableau d'octets. Généralement, les applications qui gèrent du texte créent la charge du message au format UTF-8. Le canal de télémétrie décrit cette charge de message en UTF-8, mais n'effectue aucune conversion de page de code. La chaîne de rubrique doit être au format UTF-8.

L'application est responsable de la conversion des données alphabétiques dans la page de code appropriée et des données numériques dans le codage numérique correct.

Le client MQTT Java dispose d'une méthode `MqttMessage.toString` pratique. La méthode traite la charge de message comme étant codée dans le jeu de caractères par défaut de la plateforme locale qui est généralement UTF-8. Elle convertit la charge en chaîne Java. Java dispose d'une méthode `Chaîne.getBytes` qui convertit une chaîne en tableau d'octets codé à l'aide du jeu de caractères par défaut de la plateforme locale. Deux programmes MQTT Java qui échangent du texte dans la charge de message, entre des plateformes ayant le même jeu de caractères par défaut, exécutent facilement et efficacement cette opération au format UTF-8.

Si le jeu de caractères par défaut de l'une des plateformes n'est pas UTF-8, les applications doivent alors établir une convention pour échanger les messages. Par exemple, le diffuseur de publications spécifie une conversion à partir d'une chaîne vers UTF-8 à l'aide de la méthode `getBytes("UTF8")`. Pour recevoir le texte d'un message, l'abonné part du principe que le message est codé avec le jeu de caractères UTF-8.

Le service de télémétrie (MQXR) décrit le codage de toutes les publications entrantes provenant des messages des clients MQTT comme étant au format UTF-8. Il définit `MQMD.CodedCharSetId` sur UTF-8 et `RFH2.CodedCharSetId` sur `MQCCSI_INHERIT` ; voir «[Intégration de MQ Telemetry aux gestionnaires de files d'attente](#)», à la page 140. Le format de la publication prend la valeur `MQFMT_NONE`, ce qui signifie qu'aucune conversion ne peut être effectuée par les canaux ou par `MQGET`.

Windows

Linux

AIX

## Performance et évolutivité de MQ Telemetry

Tenez compte des éléments suivants lorsque vous gérez un grand nombre de clients et que vous voulez améliorer l'évolutivité de MQ Telemetry.

### Planification de la capacité

Pour des informations sur les rapports de performances pour MQ Telemetry, voir [MQ Performance documents](#).

### Connexions

Les coûts induits par la connexion comprennent :

- le coût de configuration d'une connexion en soi, en terme d'utilisation et de temps du processeur ;
- les coûts liés au réseau ;
- la mémoire utilisée lorsqu'une connexion est ouverte et qu'elle n'est pas utilisée.

Une charge supplémentaire est à prévoir lorsque les clients restent connectés. Si une connexion est maintenue ouverte, les flux TCP/IP et les messages MQTT utilisent le réseau pour vérifier que la connexion est toujours présente. De plus, de la mémoire est utilisée au niveau du serveur pour chaque connexion client restée ouverte.

Si vous envoyez des messages à une fréquence supérieure à un message par minute, laissez votre connexion ouverte pour éviter le coût lié à l'établissement d'une nouvelle connexion. Si vous envoyez des messages à une fréquence inférieure à un message toutes les 10 ou 15 minutes, pensez à mettre fin à votre connexion afin d'éviter le coût engendré par son maintien. Il peut être préférable de conserver une connexion TLS ouverte, mais inactive, pendant de longues périodes car la configuration d'une telle connexion est plus onéreuse.

De plus, tenez compte de la capacité du client. S'il existe une fonction de stockage et retransmission sur le client, vous pouvez envoyer les messages par lots et interrompre la connexion entre l'envoi des lots. Toutefois, si le client est déconnecté, il ne peut pas recevoir de messages provenant du serveur. Vous prendrez donc votre décision en fonction de l'utilisation que vous faites de votre application.

Si, dans votre système, il existe un client qui envoie beaucoup de messages, par exemple des transferts de fichiers, alors n'attendez pas une réponse du serveur par message. Envoyez tous les messages et vérifiez à la fin de l'opération qu'ils ont tous été reçus. Vous pouvez également utiliser la qualité de service (QoS).

Vous pouvez modifier la qualité de service par message, en transmettant les messages non importants avec une qualité de service de 0 et les messages importants avec une qualité de service de 2. Le débit des messages peut être environ deux fois plus élevé avec une qualité de service de 0 qu'avec une qualité de service de 2.

## désignation des conventions

Si votre application a été conçue pour un grand nombre de clients, implémentez une convention de dénomination efficace. Pour mapper chaque client à l'identificateur de client correct, choisissez un identificateur de client significatif. Une bonne convention de dénomination permet à l'administrateur de déterminer plus facilement les clients qui sont en cours d'exécution. Elle aide l'administrateur à filtrer la longue liste des clients dans IBM MQ Explorer, et facilite l'identification des problèmes. Voir Identificateur de client.

## Débit

La longueur des noms de rubrique affecte le nombre d'octets transmis via le réseau. Lorsque vous diffusez une publication ou que vous vous abonnez à un message, le nombre d'octets dans un message peut s'avérer important. Par conséquent, limitez le nombre de caractères du nom de rubrique. Lorsqu'un client MQTT s'abonne à une rubrique, IBM MQ lui attribue un nom au format suivant :

```
ClientIdentifier: TopicName
```

Pour afficher tous les abonnements d'un client MQTT, vous pouvez utiliser la commande IBM MQ MQSC **DISPLAY** :

```
DISPLAY SUB(' ClientID1:*')
```

## Définition des ressources dans IBM MQ à utiliser par les clients MQTT

Un client MQTT se connecte à un gestionnaire de files d'attente éloignées IBM MQ. Il existe deux méthodes de base pour qu'une application IBM MQ envoie des messages à un client MQTT : définissez la file d'attente de transmission par défaut sur `SYSTEM.MQTT.TRANSMIT.QUEUE` ou utilisez des alias de gestionnaire de files d'attente. S'il existe un grand nombre de clients MQTT, définissez la file d'attente de transmission par défaut d'un gestionnaire de files d'attente. Lorsque vous utilisez le paramètre de file

d'attente de transmission par défaut, les tâches d'administration sont plus simples. Voir [Configuration des files d'attente réparties pour envoyer des messages aux clients MQTT](#).

## Amélioration de l'évolutivité en évitant les abonnements

Lorsqu'un client MQTT version 3 s'abonne à une rubrique, un abonnement est créé par le service de télémétrie (MQXR) dans IBM MQ. L'abonnement achemine les publications du client vers SYSTEM.MQTT.TRANSMIT.QUEUE. Le nom du gestionnaire de files d'attente éloignées dans l'en-tête de transmission de chaque publication correspond à l'identificateur ClientIdentifier du client MQTT qui a effectué l'abonnement. S'il existe de nombreux clients, chacun effectuant ses propres abonnements, de nombreux abonnements proxy sont gérés dans le cluster ou la hiérarchie de publication/abonnement IBM MQ. Pour plus d'informations sur l'utilisation d'une solution point-à-point plutôt que d'une solution de publication/abonnement, voir [Envoi d'un message à un client directement](#).

## Gestion d'un grand nombre de clients

Si vous souhaitez qu'un grand nombre de clients soient connectés simultanément, vous devez accroître la taille de la mémoire disponible pour le service de télémétrie (MQXR) en définissant les paramètres de la machine virtuelle Java `-Xms` et `-Xmx`. Effectuez les étapes suivantes :

1. Recherchez le fichier `java.properties` dans le répertoire de configuration du service de télémétrie ; voir [Répertoire de configuration du service de télémétrie \(MQXR\) sous Windows](#) ou [Répertoire de configuration du service de télémétrie sous Linux](#).
2. Suivez les instructions indiquées dans le fichier. Un segment de mémoire de 1 Go est suffisant pour 50 000 clients connectés simultanément.

```
# Heap sizing options - uncomment the following lines to set the heap to 1G
#-Xmx1024m
#-Xms1024m
```

3. Ajoutez d'autres arguments de ligne de commande à transmettre à la JVM exécutant le service de télémétrie (MQXR) dans le fichier `java.properties` ; voir [Transmission des paramètres JVM au service de télémétrie \(MQXR\)](#).

Pour augmenter le nombre de descripteurs de fichiers ouverts sous Linux, ajoutez les lignes suivantes à `/etc/security/limits.conf/` et connectez-vous à nouveau.

```
@mqm soft nofile 65000
@mqm hard nofile 65000
```

Chaque connecteur nécessite un descripteur de fichier. Le service de télémétrie requiert des descripteurs de fichier supplémentaires. Ce nombre doit donc être supérieur au nombre de connecteurs ouverts requis.

Le gestionnaire de files d'attente utilise un descripteur d'objet pour chaque abonnement non durable. Pour prendre en charge plusieurs abonnements non durables actifs, augmentez le nombre maximal de descripteurs actifs dans le gestionnaire de files d'attente. Par exemple :

```
echo ALTER QMGR MAXHANDS(99999999) | runmqsc qMgrName
```

Figure 49. Modification du nombre maximum de descripteurs sur Windows

```
echo "ALTER QMGR MAXHANDS(99999999)" | runmqsc qMgrName
```

Figure 50. Modification du nombre maximum de descripteurs sur Linux

## Autres considérations

Lorsque vous planifiez la configuration requise, tenez compte de la durée nécessaire au redémarrage du système. Le temps planifié d'indisponibilité peut avoir des implications sur le nombre de messages en attente de traitement. Configurez le système de telle sorte que les messages puissent être traités dans un laps de temps acceptable. Vérifiez l'espace disque, la mémoire et la puissance du processeur. Dans certaines applications client, il peut être possible de supprimer ces messages lorsque le client se reconnecte. Pour les supprimer, définissez `CleanSession` dans les paramètres de connexion du client (voir [Sessions propres](#)). Vous pouvez également publier et abonner en utilisant la meilleure qualité de service, `0`, dans un client MQTT. Voir [Qualité de service](#). Utilisez des messages non persistants pour envoyer des messages depuis IBM MQ. Les messages associés à ces qualités de service ne sont pas récupérés lorsque le système ou la connexion redémarre.

Windows

Linux

AIX

## Périphériques compatibles avec MQ Telemetry

Les clients MQTT peuvent s'exécuter sur un large éventail de périphériques, allant des détecteurs et actionneurs aux dispositifs portatifs, en passant par les systèmes embarqués dans des véhicules.

Les clients MQTT sont petits et s'exécutent sur des périphériques avec une faible capacité de mémoire et ayant une puissance de traitement limitée. Le MQTT protocol est fiable et contient des petites en-têtes qui sont adaptés aux réseaux à faible bande passante, coûteux et dont la disponibilité est intermittente.

MQ Telemetry communique avec les périphériques de télémétrie via les applications client MQTT. Ces applications utilisent les ressources suivantes qui implémentent toutes le protocole MQTT version 3 :

- Bibliothèques client suivantes :
  - Le *MQTT client for Java*, qui est utilisé pour générer des applications natives pour (par exemple) les périphériques Android, OS X, Linux ou Windows. Les applications qui utilisent cette bibliothèque client peuvent s'exécuter sur toutes les variantes de Java de la plus petite configuration CLDC (Connected Limited Device Configuration)/MIDP (Mobile Information Device Profile) à CDC (Connected Device Configuration)/Foundation, J2SE (Java Platform, Standard Edition) et J2EE (Java Platform, Enterprise Edition). La bibliothèque de classe personnalisée IBM `jclRM` est également prise en charge. La plateforme Java ME est généralement utilisée sur les petits périphériques, tels que les actionneurs, les détecteurs, les téléphones mobiles et d'autres services intégrés. La plateforme Java SE est généralement installée sur des appareils haut de gamme et intégrés, tels que les ordinateurs de bureau et les serveurs.
  - *MQTT client for C*, qui est utilisé pour construire des applications natives pour (par exemple) les unités iOS, OS X, Linux ou Windows. Cette bibliothèque client fournit une implémentation de référence C avec le client natif préconfiguré pour les systèmes Windows et Linux. L'implémentation de référence C permet à MQTT d'être porté sur une large gamme de périphériques et de plateformes. Certains systèmes Windows sur Intel, notamment Windows 7, RedHat, Ubuntu, et certains systèmes Linux sur les plateformes ARM, tels qu'Eurotech Viper, implémentent des versions de Linux qui exécutent le client C, mais IBM ne fournit pas de support pour les plateformes. Vous devez reproduire les problèmes avec le client sur une plateforme prise en charge si vous envisagez d'appeler votre centre de support IBM.
  - Le *MQTT client for Java*, qui est utilisé pour la génération d'applications basées sur un navigateur.

Les bibliothèques client MQTT sont disponibles gratuitement sur Eclipse Paho et [MQTT.org](#). Voir [IBM MQ Telemetry Transport sample programs](#).

## Sécurité dans IBM MQ

Dans IBM MQ, il existe plusieurs méthodes de sécurité : interface de service d'autorisation, exits de canal tiers ou écrits par l'utilisateur, sécurité de canal à l'aide du protocole TLS (Transport Layer Security), enregistrements d'authentification de canal et sécurité des messages.

## Interface de service d'autorisation

L'autorisation de l'utilisation d'appels MQI, de commandes et d'accès aux objets est fournie par le **gestionnaire des droits d'accès aux objets** (OAM), qui est activé par défaut. L'accès aux entités IBM MQ est contrôlé via des groupes d'utilisateurs IBM MQ et le gestionnaire OAM. Les administrateurs peuvent utiliser une interface de ligne de commande pour octroyer ou révoquer des autorisations, selon les besoins.

Pour plus d'informations sur la création de composants de service d'autorisation, voir [Configuration de la sécurité sur les systèmes AIX, Linux, and Windows](#).

## Exits de canal tiers ou écrits par l'utilisateur

Les canaux peuvent utiliser des exits de canal tiers ou écrits par l'utilisateur. Pour plus d'informations, voir [Programmes d'exit de canal pour les canaux de messagerie](#).

## Sécurité de canal à l'aide de TLS

Le protocole TLS (Transport Layer Security) fournit une sécurité de canal standard assurant la protection contre les interceptions électroniques, les falsifications et l'usurpation.

TLS utilise des techniques de clé publique et symétrique pour assurer la confidentialité et l'intégrité des messages et l'authentification mutuelle.

Pour plus de détails sur la sécurité dans IBM MQ, y compris des informations détaillées sur TLS, voir [Sécurisation](#). Pour une présentation de TLS, notamment des pointeurs des commandes décrites dans la présente section, voir [Protocoles de sécurité cryptographiques TLS](#).

## Enregistrements d'authentification de canal

Les enregistrements d'authentification de canal permettent d'exercer un contrôle précis sur l'accès octroyé aux systèmes de connexion au niveau d'un canal. Pour plus d'informations, voir [Enregistrements d'authentification de canal](#).

## Sécurité des messages

Utilisez Advanced Message Security, qui est un composant sous licence d'IBM MQ installé séparément, pour fournir une protection cryptographique aux messages envoyés et reçus via IBM MQ. Voir [Advanced Message Security](#).

### Tâches associées

[Sécurisation](#)

[Planification de la sécurité](#)

## Prise en charge de TLS par le client géré IBM MQ.NET

Le client entièrement géré IBM MQ.NET fournit la prise en charge TLS (Transport Layer Security) basée sur le kit Microsoft.NET SSLStreams. Les autres clients IBM MQ sont basés sur GSKit.

Vous pouvez développer des applications IBM MQ.NET pour les exécuter en mode géré ou non géré.

- En mode géré, les applications .NET fonctionnent dans .NET CLR (Common Language Runtime) sans aucun appel interplateforme, tel que l'appel C MQI.
- En mode non géré, C MQI est appelé pour les opérations MQI sous-jacentes. Généralement, l'interface en mode non géré comprend des classes d'encapsuleur .NET en plus de C MQI.

Le client géré IBM MQ.NET utilise les bibliothèques Microsoft.NET Framework pour implémenter les protocoles de connexion sécurisée TLS. La classe System.NET.Security.SSLStream de Microsoft est utilisée pour implémenter la sécurité (TLS) dans IBM MQ.NET.

Le mode client IBM MQ.NET non géré prend déjà en charge la fonction TLS basée sur C MQI (et GSKit). Les opérations TLS sont donc traitées par C MQI. Dans ce cas, GSKit implémente les protocoles de connexion sécurisée TLS.

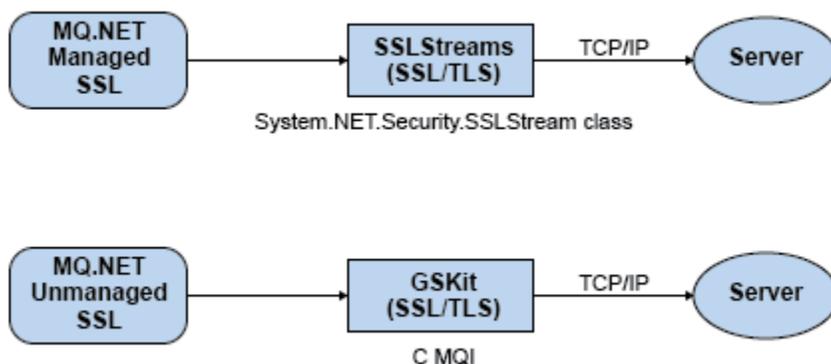


Figure 51. Comparaison de SSL/TLS géré et non géré par IBM MQ.NET

Le tableau suivant présente de manière synthétique les différences entre les implémentations gérées et non gérées :

Tableau 14. Différences entre les implémentations gérées et non gérées

Mode	Protocoles	Implémentation	Commentaires
SSL géré IBM MQ.NET	TLS	Classe System.NET.Security.SSLStream La classe SSLStream joue le rôle de flux via un socket TCP connecté	TLS 1.0 TLS 1.2 (avec Microsoft.NET Framework version 4.5 uniquement)
SSL non géré IBM MQ.NET	TLS	GSKIT et C-MQI	Protocoles de connexion sécurisée TLS

### Concepts associés

[Support Secure Sockets Layer \(SSL\) et Transport Layer Security \(TLS\) pour .NET](#)

## Clients et serveurs

Présentation de la façon dont IBM MQ prend en charge les configurations client-serveur pour ses applications.

Un *client* IBM MQ MQI est un composant qui permet à une application exécutée sur un système d'émettre des appels MQI à un gestionnaire de files d'attente exécuté sur un autre système. Le résultat de l'appel est renvoyé au client, qui le renvoie à l'application.

Un IBM MQ serveur est un gestionnaire de files d'attente qui fournit des services de mise en file d'attente à un ou plusieurs clients. Tous les objets IBM MQ, par exemple les files d'attente, existent uniquement sur le poste du gestionnaire de files d'attente (le poste serveur IBM MQ) et non sur le client. Un serveur IBM MQ peut également prendre en charge les applications IBM MQ.

La différence entre un serveur IBM MQ et un gestionnaire de files d'attente ordinaire réside dans le fait qu'un serveur possède une liaison de communication dédiée avec chaque client. Pour plus d'informations sur la création de canaux pour les clients et les serveurs, voir [Configuration de la mise en file d'attente répartie](#).

Pour plus d'informations sur les clients en général, voir «Présentation des IBM MQ MQI clients», à la page 151.

## Applications IBM MQ dans un environnement client-serveur

Une fois liées à un serveur, les applications IBM MQ client peuvent émettre la plupart des appels MQI de la même façon que les applications locales. L'application client émet un appel MQCONN pour se connecter à un gestionnaire de files d'attente spécifié. Tous les autres appels MQI spécifiant le descripteur de connexion renvoyé par la demande de connexion sont ensuite traités par ce gestionnaire de files d'attente.

Vous devez connecter vos applications aux bibliothèques client appropriées. Voir [Génération d'applications pour les IBM MQ MQI clients](#).

### Concepts associés

«Gestion et prise en charge des transactions», à la page 158

Présentation de la gestion des transactions et description de la prise en charge des transactions par IBM MQ.

«Extension des fonctions du gestionnaire de files d'attente», à la page 159

Vous pouvez étendre les fonctions du gestionnaire de files d'attente à l'aide d'exits utilisateur, d'exits API ou de services optionnels.

## Présentation des IBM MQ MQI clients

Un IBM MQ MQI client est un composant du produit IBM MQ qui peut être installé sur un système sur lequel aucun gestionnaire de files d'attente n'est exécuté.

En utilisant un IBM MQ MQI client, une application exécutée sur le même système que le client peut se connecter à un gestionnaire de files d'attente en cours d'exécution sur un autre système. L'application peut émettre des appels MQI à ce gestionnaire de files d'attente. Une application de ce type est appelée application IBM MQ MQI client et le gestionnaire de files d'attente est appelé *gestionnaire de files d'attente du serveur*.

Une application IBM MQ MQI client et un gestionnaire de files d'attente du serveur communiquent ensemble à l'aide d'un *canal MQI*. Ce dernier démarre lorsque l'application client émet un appel **MQCONN** ou **MQCONNX** pour se connecter au gestionnaire de files d'attente et s'arrête lorsque l'application client émet un appel **MQDISC** pour se déconnecter du gestionnaire de files d'attente. Les paramètres d'entrée d'un appel MQI sont transmis dans une direction sur un canal MQI et les paramètres de sortie dans la direction opposée.

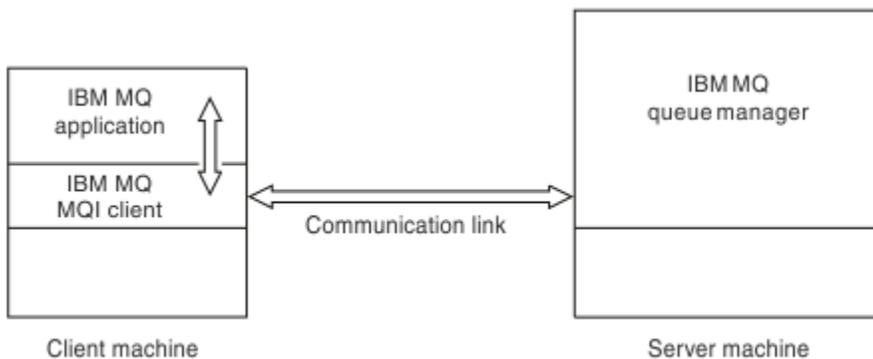


Figure 52. Liaison entre un client et un serveur

Les plateformes ci-dessous peuvent être utilisées. Les combinaisons sont fonction du produit IBM MQ que vous utilisez et sont décrites dans la rubrique [«Plateformes prises en charge pour les clients IBM MQ»](#), à la page 154.

## IBM MQ MQI client

AIX and Linux  
Windows  
IBM i

## IBM MQServeur

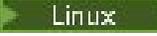
AIX and Linux  
Windows  
IBM i  
z/OS

Le canal MQI est accessible aux applications exécutées sur la plateforme client ; les files d'attente et autres objets IBM MQ sont conservés dans un gestionnaire de files d'attente que vous avez installé sur un serveur.

Une application que vous souhaitez exécuter dans l'environnement IBM MQ MQI client doit auparavant être connectée à la bibliothèque client appropriée. Lorsque l'application émet un appel MQI, le IBM MQ MQI client achemine la demande vers un gestionnaire de files d'attente dans lequel elle est traitée et à partir duquel une réponse est renvoyée au IBM MQ MQI client.

La liaison entre l'application et le IBM MQ MQI client est établie de façon dynamique au moment de l'exécution.

Vous pouvez également développer des applications client à l'aide de IBM MQ classes for .NET, IBM MQ classes for Java ou IBM MQ classes for Java Message Service (JMS). Vous pouvez utiliser les clients Java et JMS sur les plateformes suivantes :

-  IBM i
-  AIX
-  Linux
-  Windows

L'utilisation de Java et JMS n'est pas décrite dans la présente rubrique. Pour plus de détails sur l'installation, la configuration et l'utilisation d'IBM MQ classes for Java et d'IBM MQ classes for JMS, voir [Utilisation d'IBM MQ classes for Java](#) et [Utilisation d'IBM MQ classes for JMS](#).

### Concepts associés

«[Pourquoi utiliser des clients IBM MQ ?](#)», à la page 152

L'utilisation de clients IBM MQ est un moyen efficace d'implémenter la messagerie et la mise en file d'attente IBM MQ.

«[Comment configurer un IBM MQ MQI client](#)», à la page 154

Pour configurer un client, suivez les instructions ci-dessous.

«[A quoi correspond un client transactionnel étendu ?](#)», à la page 155

Un client transactionnel étendu IBM MQ peut mettre à jour les ressources gérées par un autre gestionnaire de ressources, sous le contrôle d'un gestionnaire de transactions externe.

«[Mode de connexion du client au serveur](#)», à la page 156

Un client se connecte à un serveur à l'aide de MQCONN ou de MQCONNX, et communique via un canal.

## Pourquoi utiliser des clients IBM MQ ?

L'utilisation de clients IBM MQ est un moyen efficace d'implémenter la messagerie et la mise en file d'attente IBM MQ.

Vous pouvez avoir une application qui utilise l'interface MQI exécutée sur une machine et le gestionnaire de files d'attente sur une machine différente (physique ou virtuelle). Voici les avantages :

- Mise en oeuvre complète d'IBM MQ sur la machine client.
- Configuration matérielle requise sur le système client réduite.

- Exigences de l'administration système réduites.
- Une application IBM MQ exécutée sur un client peut se connecter à plusieurs gestionnaires de files d'attente sur des systèmes différents.
- Des canaux secondaires utilisant des protocoles de transmission différents peuvent être utilisés.

### Référence associée

«Quelles applications s'exécutent sur un IBM MQ MQI client ?», à la page 153

L'interface MQI complète est prise en charge dans l'environnement client.

«Plateformes prises en charge pour les clients IBM MQ», à la page 154

IBM MQ sur toutes les plateformes serveur prises en charge accepte les connexions des IBM MQ MQI clients sur un certain nombre de plateformes.

### Quelles applications s'exécutent sur un IBM MQ MQI client ?

L'interface MQI complète est prise en charge dans l'environnement client.

Vous pouvez configurer pratiquement toutes les applications IBM MQ en vue de leur exécution sur un système IBM MQ MQI client en liant l'application sur le IBM MQ MQI client à la bibliothèque MQIC au lieu de la lier à la bibliothèque MQI. Voici les exceptions :

- MQGET avec signal
- Une application nécessitant la coordination de point de synchronisation avec d'autres gestionnaires de ressources doit utiliser un client transactionnel étendu

Si la lecture anticipée est activée pour améliorer les performances de la messagerie non persistante, les options MQGET ne sont pas toutes disponibles. Le tableau présente les options admises et indique si elles peuvent être modifiées d'un appel MQGET à l'autre.

Valeurs	Admise lorsque la lecture anticipée est activée, et modifiable d'un appel MQGET à l'autre	Admise lorsque la lecture anticipée est activée, mais non modifiable entre les appels MQGET <sup>1</sup>	Options MQGET non autorisées lorsque la lecture anticipée est activée <sup>2</sup>
Valeurs MQGET MD	MsgId <sup>3</sup> CorrelId <sup>3</sup>	Codage CodedCharSetId	
Options MQGET MQGMO	MQGMO_WAIT MQGMO_NO_WAIT MQGMO_FAIL_IF QUIESCING MQGMO_BROWSE_FIRST <sup>4</sup> MQGMO_BROWSE_NEXT <sup>4</sup> MQGMO_BROWSE_MESSAGE_UNDER_CURSOR <sup>4</sup>	MQGMO_SYNCPOINT_IF_PERSISTENT MQGMO_NO_SYNCPOINT MQGMO_ACCEPT_TRUNCATED_MSG MQGMO_CONVERT MQGMO_LOGICAL_ORDER MQGMO_COMPLETE_MSG MQGMO_ALL_MSGS_AVAILABLE MQGMO_ALL_SEGMENTS_AVAILABLE MQGMO_MARK_BROWSE_HANDLE MQGMO_MARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_CO_OP MQGMO_UNMARK_BROWSE_HANDLE MQGMO_UNMARKED_BROWSE_MSG MQGMO_PROPERTIES_FORCE_MQR FH2 MQGMO_NO_PROPERTIES MQGMO_PROPERTIES_IN_HANDLE MQGMO_PROPERTIES_COMPATIBILITY	MQGMO_SET_SIGNAL MQGMO_SYNCPOINT MQGMO_MARK_SKIP_BACKOUT MQGMO_MSG_UNDER_CURSOR <sup>4</sup> MQGMO_LOCK MQGMO_UNLOCK
Valeurs MQGMO		MsgHandle	

1. Si ces options sont modifiées d'un appel MQGET à l'autre, un code anomalie MQRC\_OPTIONS\_CHANGED est renvoyé.

2. Si ces options sont spécifiées lors du premier appel MQGET, la lecture anticipée est désactivée. Si ces options sont spécifiées lors d'un appel MQGET ultérieur, un code anomalie MQRC\_OPTIONS\_ERROR est renvoyé.
3. Les applications client doivent être informées que, si les valeurs MsgId et CorrelId sont modifiées d'un appel MQGET à l'autre, il se peut que les messages comportant les valeurs précédentes aient déjà été envoyés au client et qu'ils restent dans la mémoire tampon de lecture anticipée du client tant qu'ils ne sont pas consommés (ou automatiquement purgés).
4. Le premier appel MQGET détermine si les messages doivent être explorés ou extraits d'une file d'attente lorsque la lecture anticipée est activée. Si l'application tente d'utiliser une combinaison d'exploration et d'extraction, un code anomalie MQRC\_OPTIONS\_CHANGED est renvoyé.
5. MQGMO\_MSG\_UNDER\_CURSOR n'est pas possible avec la lecture anticipée. Les messages peuvent être explorés ou extraits lorsque la lecture anticipée est activée mais pas une combinaison des deux.

Une application exécutée sur un IBM MQ MQI client peut se connecter simultanément à plusieurs gestionnaires de files d'attente ou utiliser un nom de gestionnaire de files d'attente comportant un astérisque (\*) sur un appel MQCONN ou MQCONNX (voir les exemples dans [Connexion des applications IBM MQ MQI client aux gestionnaires de files d'attente](#)).

### **Plateformes prises en charge pour les clients IBM MQ**

IBM MQ sur toutes les plateformes serveur prises en charge accepte les connexions des IBM MQ MQI clients sur un certain nombre de plateformes.

IBM MQ installé sous la forme d'un *serveur et produit de base* sur toutes les plateformes serveur prises en charge peut accepter les connexions des IBM MQ MQI clients sur les plateformes suivantes :

-  IBM i
-  AIX
-  Linux
-  Windows

L'identificateur du jeu de caractères codés (CCSID) et le protocole de communication peuvent varier selon les connexions.

### **Comment configurer un IBM MQ MQI client**

Pour configurer un client, suivez les instructions ci-dessous.

Pour configurer un IBM MQ MQI client, vous devez avoir installé un serveur IBM MQ opérationnel auquel le client se connectera. Les étapes de la configuration d'un client sont les suivantes :

1. Vérifiez que vous disposez d'une plateforme appropriée pour un IBM MQ MQI client et que la configuration matérielle et logicielle requise est respectée. Les plateformes prises en charges sont décrites dans la rubrique «[Plateformes prises en charge pour les clients IBM MQ](#)», à la page 154.
2. Déterminez la façon dont vous allez installer IBM MQ sur votre poste de travail client, puis suivez les instructions correspondant à votre combinaison particulière de plateformes client et serveur. L'installation est décrite dans les rubriques suivantes :
  -  [Installation d'un client IBM MQ sous AIX](#)
  -  [Installation d'un client IBM MQ sous Linux](#)
  -  [Installation d'un client IBM MQ sous Windows](#)
  -  [Installation d'un client IBM MQ sous IBM i](#)
3. Assurez-vous que vos liaisons de communication sont configurées et connectées. La configuration des liaisons de communication est décrite dans la rubrique [Configuration des connexions entre le serveur et le client](#).

4. Vérifiez que votre installation fonctionne correctement. Consultez la section de vérification de la procédure d'installation pour la ou les plateformes que votre entreprise utilise.
5. Une fois que vous avez vérifié l'installation du IBM MQ MQI client, déterminez si vous devez sécuriser le client. La sécurité du client est décrite dans [Configuration de la sécurité du IBM MQ MQI client](#).
6. Configurez les canaux entre le IBM MQ MQI client et le serveur, qui sont nécessaires aux applications IBM MQ à exécuter sur le client. La configuration des canaux est décrite dans la rubrique [Définition de canaux MQI](#). Si vous utilisez TLS, vous devez prendre en compte quelques informations supplémentaires. Ces informations sont décrites dans la rubrique [Spécification de l'utilisation de TLS par un canal MQI](#). Il peut être nécessaire d'utiliser un fichier de configuration IBM MQ MQI client ou des variables d'environnement IBM MQ pour configurer les canaux. Les variables d'environnement IBM MQ sont décrites dans [Utilisation des variables d'environnement IBM MQ](#).
7. Les applications IBM MQ sont décrites en intégralité dans [Développement d'applications](#).
8. Vous devez prendre en considération certaines différences par rapport à un environnement de gestionnaire de files d'attente lors de la conception, de la génération et de l'exécution d'applications dans l'environnement du IBM MQ MQI client. Pour plus d'informations sur ces différences, voir :
  - [Utilisation de l'interface de file d'attente de messages \(MQI\) dans une application client](#)
  - [Développement d'applications pour les IBM MQ MQI clients](#)
  - [Connexion des applications de IBM MQ MQI client aux gestionnaires de files d'attente](#)
  - [Résolution des incidents liés aux IBM MQ MQI clients](#)

## A quoi correspond un client transactionnel étendu ?

Un client transactionnel étendu IBM MQ peut mettre à jour les ressources gérées par un autre gestionnaire de ressources, sous le contrôle d'un gestionnaire de transactions externe.

Si vous ne maîtrisez pas les concepts de la gestion des transactions, consultez la rubrique [«Gestion et prise en charge des transactions»](#), à la page 158.

Notez que le client transactionnel XA est maintenant fourni avec IBM MQ.

Une application client peut participer à une unité d'oeuvre gérée par un gestionnaire de files d'attente auquel elle est connectée. Au sein de l'unité d'oeuvre, l'application client peut placer des messages dans des files d'attente (et d'en extraire) appartenant à ce gestionnaire de files d'attente. L'application client peut ensuite utiliser l'appel **MQCMIT** pour valider l'unité d'oeuvre ou l'appel **MQBACK** pour annuler l'unité d'oeuvre. Cependant, au sein de la même unité d'oeuvre, l'application client ne peut pas mettre à jour les ressources d'un autre gestionnaire de ressources, par exemple les tables d'une base de données Db2. L'utilisation d'un client transactionnel étendu IBM MQ permet de supprimer cette restriction.

Un client transactionnel étendu IBM MQ est un IBM MQ MQI client doté d'une fonction supplémentaire. Cette fonction permet à une application client, au sein de la même unité d'oeuvre :

- de placer des messages dans des files d'attente (et d'en extraire) appartenant au gestionnaire de files d'attente auquel elle est connectée ;
- de mettre à jour les ressources d'un gestionnaire de ressources autre qu'un gestionnaire de files d'attente IBM MQ

Cette unité d'oeuvre doit être gérée par un gestionnaire de transactions externe exécuté sur le même système que l'application client. L'unité d'oeuvre ne peut pas être gérée par le gestionnaire de files d'attente auquel l'application client est connectée. En d'autres termes, le gestionnaire de files d'attente peut faire office uniquement de gestionnaire de ressources, et non de gestionnaire de transactions. Cela signifie également que l'application client peut valider ou annuler l'unité d'oeuvre à l'aide de l'interface de programmation d'application (API) fournie par le gestionnaire de transactions externe. L'application client ne peut donc pas utiliser les appels MQI, **MQBEGIN**, **MQCMIT** et **MQBACK**.

Le gestionnaire de transactions externe communique avec le gestionnaire de files d'attente en tant que gestionnaire de ressources à l'aide du même canal MQI que celui utilisé par l'application client connectée au gestionnaire de files d'attente. Cependant, dans une situation de reprise après incident, lorsqu'aucune application n'est en cours d'exécution, le gestionnaire de transactions peut utiliser un canal MQI dédié

pour rétablir les unités d'oeuvre incomplètes auxquelles participait le gestionnaire de files d'attente au moment de l'incident.

Dans cette section, un IBM MQ MQI client qui ne possède pas la fonction transactionnelle étendue s'appelle un client de base IBM MQ. Vous pouvez donc considérer qu'un client transactionnel étendu IBM MQ est constitué d'un client de base IBM MQ auquel s'ajoute la fonction transactionnelle étendue.

**Remarque :**  Le IBM MQ MQI client sur IBM i ne prend pas en charge la fonction transactionnelle étendue IBM MQ.

### Référence associée

«Plateformes prises en charge pour les clients transactionnels étendus», à la page 156

Les clients transactionnels étendus sont disponibles pour toutes les éditions Multiplatforms prenant en charge un client de base. Les clients ne sont pas disponibles pour z/OS.

### **Plateformes prises en charge pour les clients transactionnels étendus**

Les clients transactionnels étendus sont disponibles pour toutes les éditions Multiplatforms prenant en charge un client de base. Les clients ne sont pas disponibles pour z/OS.

Une application client qui utilise un client transactionnel étendu peut se connecter à un gestionnaire de files d'attente des produits IBM MQ 9.0 suivants uniquement :

-  IBM MQ for AIX
-  IBM MQ for IBM i
-  IBM MQ pour Linux
-  IBM MQ for Windows

 Bien qu'il n'existe aucun client transactionnel étendu qui s'exécute sous z/OS, une application client utilisant un client de ce type peut se connecter à un gestionnaire de files d'attente qui s'exécute sous z/OS.

Pour chaque plateforme, la configuration matérielle et logicielle requise pour le client transactionnel étendu est la même que celle du client de base IBM MQ. Un langage de programmation est pris en charge par un client transactionnel étendu s'il est pris en charge par le client de base IBM MQ et par le gestionnaire de files d'attente que vous utilisez.

Pour plus d'informations sur les gestionnaires de transactions externes pour toutes les plateformes, voir [Configuration système requise pour IBM MQ](#).

## Mode de connexion du client au serveur

Un client se connecte à un serveur à l'aide de MQCONN ou de MQCONNX, et communique via un canal.

Une application exécutée dans l'environnement client IBM MQ doit maintenir une connexion active entre les postes client et serveur.

La connexion est établie par une application émettant un appel MQCONN ou MQCONNX. Les clients et les serveurs communiquent via des *canaux MQI* ou, lors de l'utilisation de conversations partagées, chacun partage une instance de canal MQI. Lorsque l'appel aboutit, l'instance de canal MQI ou la conversation reste connectée tant que l'application n'émet pas un appel MQDISC. C'est le cas pour tous les gestionnaires de files d'attente auxquels une application doit se connecter.

### Concepts associés

«Client et gestionnaire de files d'attente sur la même machine», à la page 157

Vous pouvez également exécuter une application dans l'environnement IBM MQ MQI client lorsque votre machine dispose également d'un gestionnaire de files d'attente installé.

«Clients sur des plateformes différentes», à la page 157

Voici un autre exemple de système IBM MQ MQI client et serveur. Dans cet exemple, la machine serveur communique avec trois IBM MQ MQI clients sur différentes plateformes.

«Utilisation des versions différentes des logiciels client et serveur», à la page 157

Si vous utilisez des versions antérieures des produits IBM MQ, assurez-vous que la conversion de code du CCSID de votre client est prise en charge par le serveur.

### **Client et gestionnaire de files d'attente sur la même machine**

Vous pouvez également exécuter une application dans l'environnement IBM MQ MQI client lorsque votre machine dispose également d'un gestionnaire de files d'attente installé.

Dans cette situation, vous avez le choix de vous connecter aux bibliothèques du gestionnaire de files d'attente ou aux bibliothèques client, mais notez que, si vous vous connectez aux bibliothèques client, vous devez néanmoins définir les connexions de canal. Cela peut être utile lors de la phase de développement d'une application. Vous pouvez tester votre programme sur votre propre machine, sans aucune dépendance, sur d'autres machines et vous assurer qu'il continue de fonctionner lorsque vous le placez dans un environnement IBM MQ MQI client indépendant.

### **Clients sur des plateformes différentes**

Voici un autre exemple de système IBM MQ MQI client et serveur. Dans cet exemple, la machine serveur communique avec trois IBM MQ MQI clients sur différentes plateformes.

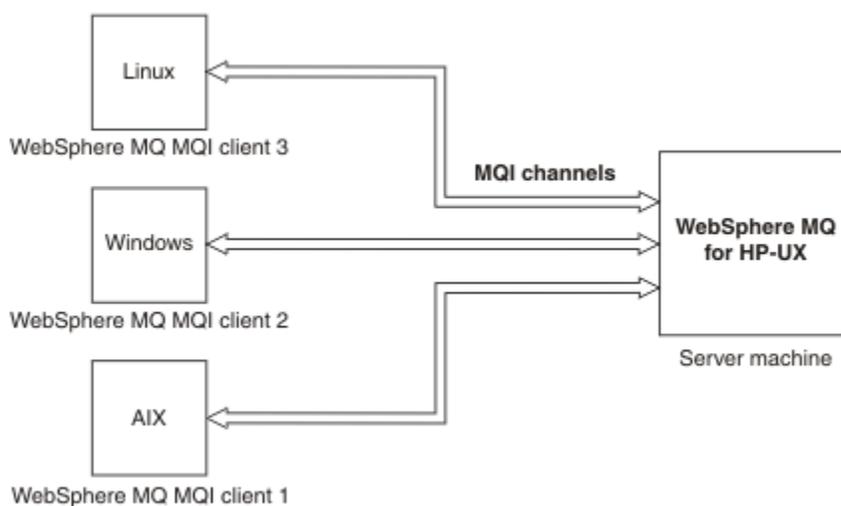


Figure 53. Serveur IBM MQ connecté à des clients sur des plateformes différentes

D'autres environnements plus complexes sont possibles. Par exemple, un client IBM MQ peut se connecter à plusieurs gestionnaires de files d'attente ou à n'importe quel nombre de gestionnaires de files d'attente connectés dans le cadre d'un groupe de partage de files d'attente.

### **Utilisation des versions différentes des logiciels client et serveur**

Si vous utilisez des versions antérieures des produits IBM MQ, assurez-vous que la conversion de code du CCSID de votre client est prise en charge par le serveur.

Un client IBM MQ peut se connecter à toutes les versions prises en charge du gestionnaire de files d'attente. Si vous vous connectez à une version antérieure du gestionnaire de files d'attente, vous ne pouvez pas utiliser les fonctions et structures d'une version antérieure du produit dans votre application IBM MQ sur le client.

Un gestionnaire de files d'attente IBM MQ peut communiquer avec des clients de différentes versions avec lui-même en négociant jusqu'au niveau de protocole le plus élevé mutuellement pris en charge. Cela signifie que les clients plus anciens peuvent être utilisés avec des niveaux de gestionnaire de files d'attente plus récents. Il est recommandé que le client et le serveur soient au niveau des versions de IBM MQ qui sont actuellement prises en charge pour faciliter le diagnostic des problèmes et activer la prise en charge par IBM.

Pour plus d'informations, voir les langages de programmation pris en charge dans [Développement d'applications](#).

## Gestion et prise en charge des transactions

---

Présentation de la gestion des transactions et description de la prise en charge des transactions par IBM MQ.

Un *gestionnaire de ressources* est un sous-système informatique détenant et gérant des ressources accessibles aux applications et mises à jour par ces dernières. Voici des exemples de gestionnaires de ressources :

- Un gestionnaire de files d'attente IBM MQ dont les files d'attente sont les ressources
- Base de données Db2 dont ses tables sont les ressources

Lorsqu'une application met à jour les ressources d'un ou de plusieurs gestionnaires de ressources, il peut y avoir un besoin métier garantissant que certaines mises à jour aboutissent toutes sous la forme d'un groupe ou qu'aucune n'aboutit. La raison de ce type de besoin est que les données métier restent dans un état incohérent si certaines de ces mises à jour ont abouti alors que d'autres n'ont pas abouti.

Les mises à jour des ressources gérées de cette façon sont censées se produire au sein d'une *unité d'oeuvre*, à savoir une *transaction*. Un programme d'application peut regrouper un ensemble de mises à jour dans une unité d'oeuvre.

Pendant une unité d'oeuvre, une application émet des demandes aux gestionnaires de ressources pour mettre à jour leurs ressources. L'unité d'oeuvre prend fin lorsque l'application émet une demande de validation de toutes les mises à jour. Tant que les mises à jour ne sont pas validées, aucune d'elles n'est visible par d'autres applications accédant aux mêmes ressources. Le cas échéant, si l'application décide qu'elle ne peut pas terminer l'unité d'oeuvre pour une raison quelle qu'elle soit, elle peut émettre une demande pour annuler toutes les mises à jour qu'elle a demandées jusqu'à ce stade. Dans ce cas, aucune mise à jour ne devient visible par d'autres applications. Ces mises à jour sont généralement associées de façon logique et doivent toutes aboutir pour que l'intégrité des données soit préservée. Si une mise à jour aboutit alors qu'une autre échoue, l'intégrité des données est perdue.

Lorsqu'une unité d'oeuvre aboutit, elle est censée effectuer une *validation*. Une fois validées, toutes les mises à jour effectuées dans cette unité d'oeuvre deviennent permanentes et irréversibles. Cependant, si l'unité d'oeuvre échoue, toutes les mises à jour sont *annulées*. Ce processus au cours duquel les unités d'oeuvre sont validées ou annulées avec intégrité est appelé *coordination de point de synchronisation*.

Le stade auquel toutes les mises à jour au sein d'une unité d'oeuvre sont validées ou annulées est appelé *point de synchronisation*. Une mise à jour au sein d'une unité d'oeuvre est censée se produire *sous le contrôle d'un point de synchronisation*. Si une application demande une mise à jour qui est *en dehors du contrôle d'un point de synchronisation*, le gestionnaire de ressources valide immédiatement la mise à jour, même si une unité d'oeuvre est en cours, et la mise à jour ne peut pas être ultérieurement.

Le sous-système informatique qui gère les unités d'oeuvre est appelé *gestionnaire de transactions* ou *coordinateur de point*.

Une unité d'oeuvre *locale* est une unité d'oeuvre dans laquelle les seules ressources mises à jour sont celles du gestionnaire de files d'attente IBM MQ. Dans ce cas, la coordination de point de synchronisation est fournie par le gestionnaire de files d'attente proprement dit, à l'aide d'un processus de validation à une seule phase.

Une unité d'oeuvre *globale* est une unité d'oeuvre dans laquelle les ressources appartenant à d'autres gestionnaires de files d'attente, tels que les bases de données compatibles XA, sont également mises à jour. Dans ce cas, une procédure de validation en deux phases doit être utilisée, et l'unité d'oeuvre peut être coordonnée par le gestionnaire de files d'attente proprement dit ou en externe par un autre gestionnaire de transactions compatible XA, tel que IBM TXSeries, ou BEA.

Un gestionnaire de transactions est chargé de garantir que toutes les mises à jour des ressources au sein d'une unité d'oeuvre aboutissent ou qu'aucune d'elles n'aboutit. Pour valider ou annuler une unité d'oeuvre, une application émet une demande à un gestionnaire de transactions. CICS et WebSphere Application Server sont des exemples de gestionnaires de transactions, bien que les deux possèdent également d'autres fonctions.

Certains gestionnaires de ressources fournissent leur propre fonction de gestion de transactions. Par exemple, un gestionnaire de files d'attente IBM MQ peut gérer des unités d'oeuvre impliquant des mises à jour de ses propres ressources et des mises à jour des tables Db2. Le gestionnaire de files d'attente n'a pas besoin d'un gestionnaire de transactions distinct pour effectuer cette fonction, même si un gestionnaire de transactions peut être utilisé s'il s'agit d'un besoin métier. Si un gestionnaire de transactions distinct est utilisé, il est désigné par *gestionnaire de transactions externe*.

Pour qu'un gestionnaire de transactions externe gère une unité d'oeuvre, une interface standard doit exister entre le gestionnaire de transactions et tous les gestionnaires de ressources participant à l'unité d'oeuvre. Cette interface permet au gestionnaire de transactions et à un gestionnaire de ressources de communiquer l'un avec l'autre. L'une de ces interfaces est l'*interface XA*, qui correspond à une interface standard prise en charge par un certain nombre de gestionnaires de transactions et de gestionnaires de ressources. L'interface XA est publiée par The Open Group dans le document *Distributed Transaction Processing: The XA Specification*.

Lorsque plusieurs gestionnaires de ressources participent à une unité d'oeuvre, un gestionnaire de transactions doit utiliser un protocole de *validation à deux phases* pour garantir que toutes les mises à jour au sein de l'unité d'oeuvre aboutissent ou qu'aucune d'elles n'aboutit, même si une erreur système se produit. Lorsqu'une application émet une demande à un gestionnaire de transactions pour valider une unité d'oeuvre, le gestionnaire de transactions effectue les opérations suivantes :

#### **Phase 1 (préparation de la validation)**

Le gestionnaire de transactions demande à chaque gestionnaire de ressources participant à l'unité d'oeuvre de garantir que toutes les informations relatives aux mises à jour prévues de ses ressources sont dans un état récupérable. Un gestionnaire de ressources effectue généralement ce processus en écrivant les informations dans un journal et en garantissant que les informations sont écrites sur disque dur. La phase 1 se termine lorsque le gestionnaire de transactions reçoit de la part de chaque gestionnaire de ressources une notification selon laquelle les informations relatives aux mises à jour prévues de ses ressources sont dans un état récupérable.

#### **Phase 2 (validation)**

Une fois la phase 1 terminée, le gestionnaire de transactions prend la décision irrévocable de valider l'unité d'oeuvre. Il demande à chaque gestionnaire de ressources participant à l'unité d'oeuvre de valider les mises à jour de ses ressources. Lorsqu'un gestionnaire de ressources reçoit cette demande, il doit valider les mises à jour. Il n'a pas la possibilité de les annuler à ce stade. La phase 2 se termine lorsque le gestionnaire de transactions reçoit de la part de chaque gestionnaire de ressources une notification selon laquelle il a validé les mises à jour de ses ressources.

L'interface XA utilise un protocole de validation à deux phases.

Pour plus d'informations, voir [Scénarios de support transactionnel](#).

IBM MQ prend également en charge Microsoft Transaction Server (COM +). [L'utilisation de Microsoft Transaction Server \(COM +\)](#) permet d'obtenir des informations sur la configuration de IBM MQ pour bénéficier du support COM +.

## **Extension des fonctions du gestionnaire de files d'attente**

---

Vous pouvez étendre les fonctions du gestionnaire de files d'attente à l'aide d'exits utilisateur, d'exits API ou de services optionnels.

### **Exits utilisateur**

Les exits utilisateur vous permettent d'insérer votre propre code dans une fonction du gestionnaire de files d'attente. Les exits utilisateur pris en charge sont les suivants :

#### **Exits de canal**

Ces exits modifient le mode de fonctionnement des canaux. Les exits de canal sont décrits dans [Programmes d'exit de canal pour les canaux de messagerie](#).

### Exits de conversion de données

Ces exits créent des fragments de code source pouvant être placés dans des programmes d'application pour convertir des données d'un format à un autre. Les exits de conversion de données sont décrits dans [Ecriture d'exits de conversion de données](#).

### Exit de charge de travail du cluster

La fonction de cet exit est définie par le fournisseur de l'exit. Des informations sur les définitions d'appel sont fournies dans [MQ\\_CLUSTER\\_WORKLOAD\\_EXIT - Description d'appel](#).

### Exits API

Les exits API permettent d'écrire du code modifiant le comportement des appels API IBM MQ, tels que MQPUT et MQGET, puis d'insérer le code immédiatement avant ou après ces appels. L'insertion est automatique ; le gestionnaire de files d'attente conduit le code d'exit aux points enregistrés. Pour plus d'informations sur les exits API, voir [Utilisation et écriture d'exits API](#).

### Services optionnels

Les services optionnels possèdent des interfaces formalisées (API) dotées de plusieurs points d'entrée.

Une implémentation d'un service optionnel est appelée *composant de service*. Vous pouvez utiliser les composants fournis avec IBM MQ, ou écrire votre propre composant pour exécuter les fonctions dont vous avez besoin.

A l'heure actuelle, les services optionnels suivants sont fournis :

#### Serveur d'autorisation

Le service d'autorisation vous permet de générer votre propre fonction de sécurité.

Le composant de service par défaut implémentant le service est le gestionnaire des droits d'accès aux objets (OAM). Par défaut, le gestionnaire des droits d'accès aux objets (OAM) est actif et vous n'avez pas à intervenir pour le configurer. Vous pouvez utiliser l'interface de service d'autorisation pour créer d'autres composants afin de remplacer ou d'étendre le gestionnaire OAM. Pour plus d'informations sur l'OAM, voir [Configuration de la sécurité sur les systèmes AIX, Linux, and Windows](#).

#### Service annuaire

Le service annuaire permet aux applications de partager des files d'attente en identifiant les files d'attente éloignées comme si elles étaient des files d'attente locales.

Vous pouvez écrire votre propre composant de service annuaire. Vous pouvez être amené à le faire si vous avez l'intention d'utiliser le service annuaire avec IBM MQ, par exemple. Pour utiliser le service annuaire, vous devez disposer d'un composant écrit par l'utilisateur ou fourni par un fournisseur de logiciel différent. Par défaut, le service annuaire est inactif.

#### Concepts associés

[Exits utilisateur, exits API et services optionnels d'IBM MQ](#)

## Interfaces de langage IBM MQ Java

---

IBM MQ fournit deux autres API à utiliser dans des applications Java : IBM MQ classes for Java Message Service et IBM MQ classes for Java.

IBM soutient les normes ouvertes et y participe activement, et, dans le domaine de la messagerie, la norme API est Java Message Service (JMS). À partir de IBM MQ 8.0, le produit implémente la norme JMS 2.0, qui introduit une nouvelle API simplifiée avec des fonctionnalités telles que des abonnements partagés. De plus, WebSphere Liberty prend en charge JMS 2.0 avec le fournisseur de messagerie par défaut et IBM MQ.

Dans IBM MQ, deux API alternatives peuvent être utilisées dans les applications Java :

#### IBM MQ classes for JMS

IBM MQ classes for Java Message Service (JMS) est le fournisseur JMS livré avec IBM MQ. Java Platform, Enterprise Edition Connector Architecture (JCA) fournit une méthode standard de connexion

des applications en cours d'exécution dans un environnement Java EE à un système d'information d'entreprise (EIS) tel qu'IBM MQ ou Db2.

### **IBM MQ classes for Java**

IBM MQ classes for Java vous permet d'utiliser IBM MQ dans un environnement Java. IBM MQ classes for Java permet à une application Java de se connecter à IBM MQ en tant que client IBM MQ ou de se connecter directement à un gestionnaire de files d'attente IBM MQ.

**Remarque :** IBM MQ classes for Java est fonctionnellement stabilisé au niveau livré dans IBM MQ 8.0. Les applications existantes qui utilisent IBM MQ classes for Java continueront d'être intégralement prises en charge, mais cette API est stabilisée, ce qui signifie qu'aucune nouvelle fonction ne sera ajoutée et que toute demande d'amélioration sera rejetée. "Intégralement prises en charge" signifie que les incidents seront corrigés et que toute modification nécessaire suite à la modification de la configuration système requise pour IBM MQ sera apportée.

A partir d'IBM MQ 8.0, IBM MQ classes for Java et IBM MQ classes for JMS sont générés avec Java 7. L'environnement d'exécution Java 7 prend en charge l'exécution de versions de fichier de classe antérieures.

### **Concepts associés**

[Pourquoi utiliser IBM MQ classes for JMS ?](#)

[Pourquoi utiliser IBM MQ classes for Java ?](#)

[Modèle JMS](#)

### **Tâches associées**

[Utilisation de la fonctionnalité JMS 2.0](#)

## **IBM MQ classes for JMS**

IBM MQ classes for JMS est le fournisseur JMS fourni avec IBM MQ. IBM MQ classes for JMS implémente les interfaces définies dans le package `javax.jms`, et fournit également deux ensembles d'extensions à l'API JMS. Les applications Java Platform, Standard Edition (Java SE) et Java Platform, Enterprise Edition (Java EE) peuvent utiliser IBM MQ classes for JMS.

La spécification JMS définit un ensemble d'interfaces que peuvent utiliser les applications pour exécuter des opérations de messagerie. Depuis IBM MQ 8.0, le produit prend en charge la version JMS 2.0 de la norme JMS. Cette implémentation fournit toutes les fonctions de l'API classique, mais requiert moins d'interfaces et elle est plus simple à utiliser. Pour plus d'informations, voir «[Modèle JMS](#)», à la page 162 et la spécification JMS 2.0 sur [Java.net](#).

Le package `javax.jms` spécifie les détails des interfaces JMS et un fournisseur JMS implémente ces interfaces pour un produit de messagerie donné. IBM MQ classes for JMS est un fournisseur JMS qui implémente les interfaces JMS pour IBM MQ et fournit également les deux ensembles d'extensions à l'API JMS :

- Extensions IBM MQ JMS
- Extensions IBM JMS

Une fabrique de connexions, une file d'attente ou un objet de rubrique créés en utilisant l'interface `javax.jms` ou l'un ou l'autre des ensembles d'extensions JMS peuvent être adressés en utilisant ces API, à savoir qu'ils peuvent être diffusés vers n'importe laquelle des interfaces. Pour conserver la portabilité des applications au plus haut niveau, utilisez l'API la plus générique qui convient à vos exigences.

### **Extensions IBM MQ JMS**

IBM MQ classes for JMS fournit également des extensions à l'API JMS. Les éditions précédentes d'IBM MQ classes for JMS contenaient des extensions implémentées dans les objets `MQConnectionFactory`, `MQQueue` et `MQTopic`. Ces objets ont des propriétés et des méthodes spécifiques à IBM MQ. Les objets peuvent être des objets administrés ou une application peut créer les objets de façon dynamique lors de la phase d'exécution. IBM MQ classes for JMS gère ces extensions, et vous pouvez continuer à employer sans modification les applications qui utilisent ces extensions. Ces extensions s'appellent également des

extensions IBM MQ JMS. Notez que dans cette documentation, les objets créés de manière dynamique par une application lors de la phase d'exécution ne sont pas considérés comme des objets gérés.

## Extensions IBM JMS

En plus des extensions IBM MQ JMS, IBM MQ classes for JMS fournit un ensemble plus générique d'extensions à l'API JMS. Il s'agit des extensions IBM JMS ; elles ont les objectifs généraux suivants :

- Fournir un niveau de cohérence plus élevé entre les fournisseurs IBM JMS
- Faciliter l'écriture d'une application de passerelle entre deux systèmes de messagerie IBM
- Faciliter le portage d'une application d'un fournisseur IBM JMS vers un autre

L'objectif principal de ces extensions concerne la création et la configuration dynamique des fabriques de connexions et des destinations lors de l'exécution, mais les extensions fournissent également des fonctions qui ne sont pas directement liées à la messagerie, telles que la fonction de détermination des incidents.

## Modèle JMS

Le modèle JMS définit un ensemble d'interfaces que les applications Java peuvent utiliser pour effectuer des opérations de messagerie. IBM MQ classes for JMS, en tant que fournisseur JMS, définit comment les objets JMS sont liés aux concepts IBM MQ. La spécification JMS s'attend à ce que certains objets JMS soient des objets administrateurs.

La spécification JMS et le package `javax.jms` définissent un ensemble d'interfaces que les applications Java peuvent utiliser pour effectuer des opérations de messagerie.

Depuis IBM MQ 8.0, le produit prend en charge la version JMS 2.0 de la norme JMS qui introduit une API simplifiée, tout en conservant l'API classique de JMS 1.1.

## API simplifiée

JMS 2.0 introduit l'API simplifiée tout en conservant les interfaces dépendant ou non du domaine issues de JMS 1.1. L'API simplifiée réduit le nombre d'objets nécessaires à l'envoi et à la réception de messages et se compose des interfaces ci-dessous :

### ConnectionFactory

ConnectionFactory est un objet géré qui est utilisé par un client JMS pour créer une connexion. Cette interface est également utilisée dans l'API classique.

### JMSContexte

Cet objet combine les objets Connection et Session de l'API classicCd. Les objets JMSContext peuvent être créés à partir d'autres objets JMSContext, auquel cas la connexion sous-jacente est dupliquée.

### JMSExpéditeur

Un objet JMSProducer est créé par un objet JMSContext et il est utilisé pour envoyer des messages à une file d'attente ou à une rubrique. L'objet JMSProducer entraîne la création des objets nécessaires à l'envoi du message.

### JMSUtilisateur

Un objet JMSConsumer est créé par un objet JMSContext et il est utilisé pour recevoir des messages provenant d'une rubrique ou d'une file d'attente.

L'API simplifiée a un certain nombre d'effets :

- L'objet JMSContext démarre toujours automatiquement la connexion sous-jacente.
- Les objets JMSProducers et JMSConsumers peuvent maintenant traiter directement le corps des messages, sans devoir détenir l'objet message complet, par le biais de la méthode `getBody` du message.
- Les propriétés du message peuvent être définies sur l'objet JMSProducer à l'aide du chaînage de méthodes avant d'envoyer un 'corps', c'est-à-dire le contenu des messages. L'objet JMSProducer traite

la création de tous les objets nécessaires à l'envoi du message. A l'aide de JMS 2.0, les propriétés sont définies et le message suivant est envoyé :

```
context.createProducer().
setProperty("foo", "bar").
setTimeToLive(10000).
setDeliveryMode(NON_PERSISTENT).
setDisableMessageTimestamp(true).
send(dataQueue, body);
```

JMS 2.0 introduit également des abonnements partagés où les messages peuvent être partagés entre plusieurs consommateurs. Tous les abonnements JMS 1.1 sont traités en tant qu'abonnements non partagés.

## API classique

Les interfaces JMS principales qui constituent l'API classique sont répertoriées ci-dessous :

### Destination

Une destination représente l'endroit où une application envoie des messages, l'endroit d'où elle en reçoit, ou les deux.

### ConnectionFactory

Un objet ConnectionFactory encapsule un ensemble de propriétés de configuration pour une connexion. Une application utilise une fabrique de connexions pour créer une connexion.

### Connexion

Un objet Connection encapsule la connexion active d'une application sur un serveur de messagerie. Une application utilise une connexion pour créer des sessions.

### Session

Une session est un contexte à unité d'exécution unique pour l'envoi et la réception de messages. Une application utilise ensuite une session pour créer des messages, des expéditeurs de message et des consommateurs de message. Une session est transactionnelle ou non transactionnelle.

### Message

Un objet Message encapsule un message envoyé ou reçu par une application.

### MessageProducer

Une application utilise un expéditeur de message pour envoyer des messages vers une destination.

### MessageConsumer

Une application utilise un consommateur de message pour la réception des messages envoyés vers une destination.

La Figure 54, à la page 163 présente ces objets et leurs relations.

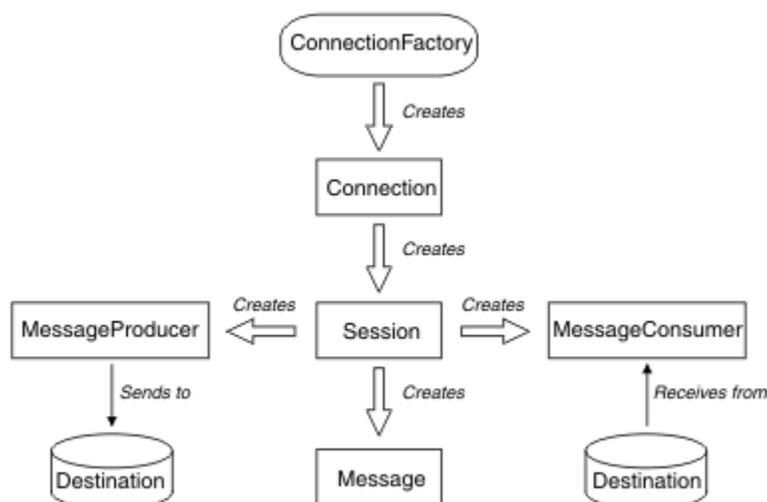


Figure 54. Objets JMS et leurs relations

Le diagramme présente les interfaces principales : ConnectionFactory, Connection, Session, MessageProducer, MessageConsumer, Message et Destination. Une application utilise une fabrique de connexions pour créer une connexion et utilise une connexion pour créer des sessions. L'application peut ensuite utiliser une session pour créer des messages, des expéditeurs de message et des consommateurs de message. L'application utilise un expéditeur de message pour envoyer des messages vers une destination et utilise un consommateur de message pour recevoir des messages envoyés vers une destination.

Un objet Destination, ConnectionFactory ou Connection peut être utilisé simultanément par différentes unités d'exécution d'une application comportant plusieurs unités d'exécution, mais un objet Session, MessageProducer ou MessageConsumer ne peut pas être utilisé par des unités d'exécution différentes. La manière la plus simple de s'assurer qu'un objet Session, MessageProducer ou MessageConsumer n'est pas utilisé simultanément consiste à créer un objet Session distinct pour chaque unité d'exécution.

JMS prend en charge deux styles de messagerie :

- Messagerie point-à-point
- Messagerie de type publication/abonnement

Ces deux styles de messagerie sont également appelés *domaines de messagerie* et ils peuvent être combinés dans une application. Dans le domaine point-à-point, une destination est une file d'attente et, dans le domaine de publication/abonnement, une destination est une rubrique.

Avec les versions de JMS antérieures à JMS 1.1, la programmation pour le domaine point-à-point utilise un ensemble d'interfaces et de méthodes et la programmation pour le domaine de publication/abonnement utilise un autre ensemble. Les deux ensembles sont similaires, mais distincts. A partir de JMS 1.1, vous pouvez utiliser un ensemble commun d'interfaces et de méthodes qui prennent en charge les deux domaines de messagerie. Les interfaces communes fournissent une vue indépendante du domaine de chaque domaine de messagerie. Le [Tableau 16](#), à la [page 164](#) affiche la liste des interfaces indépendantes du domaine JMS et leurs interfaces propres au domaine correspondantes.

*Tableau 16. Les interfaces indépendantes du domaine JMS et leurs interfaces propres au domaine correspondantes*

<b>Interfaces indépendantes du domaine</b>	<b>Interfaces propres au domaine pour le domaine point-à-point</b>	<b>Interfaces propres au domaine pour le domaine de publication/abonnement</b>
ConnectionFactory	QueueConnectionFactory	TopicConnectionFactory
Connexion	QueueConnection	TopicConnection
Destination	File d'attente	Topic
Session	QueueSession	TopicSession
MessageProducer	QueueSender	TopicPublisher
MessageConsumer	QueueReceiver QueueBrowser	TopicSubscriber

JMS 2.0 conserve toutes les interfaces propres au domaine qui peuvent donc encore être utilisées par les applications existantes. Pour les nouvelles applications, cependant, envisagez d'utiliser les interfaces indépendantes du domaine de JMS 1.1 ou l'API simplifiée de JMS 2.0.

Dans IBM MQ classes for JMS, les objets JMS sont liés aux concepts IBM MQ de plusieurs manières :

- Un objet Connection possède des propriétés issues des propriétés de la fabrique de connexions qui a été utilisée pour créer la connexion. Ces propriétés contrôlent comment une application se connecte à un gestionnaire de files d'attente. Des exemples de ces propriétés sont le nom du gestionnaire de files d'attente et, pour une application qui se connecte au gestionnaire de files d'attente en mode client, le nom d'hôte ou l'adresse IP du système sur lequel s'exécute le gestionnaire de files d'attente.

- Un objet Session encapsule le descripteur de connexion IBM MQ qui définit la portée transactionnelle de la session.
- Un objet MessageProducer et un objet MessageConsumer encapsule chacun un descripteur d'objet IBM MQ.

Lors de l'utilisation d'IBM MQ classes for JMS, toutes les règles normales d'IBM MQ s'appliquent. Sachez, en particulier, qu'une application peut envoyer un message à une file d'attente éloignée, mais elle peut uniquement recevoir un message d'une file d'attente qui appartient au gestionnaire de files d'attente auquel l'application est connectée.

La spécification JMS s'attend à ce que les objets ConnectionFactory et Destination soient des objets gérés. Un administrateur crée et gère les objets gérés dans un référentiel central et une application JMS extrait ces objets à l'aide de l'interface JNDI (Java Naming and Directory Interface).

Dans IBM MQ classes for JMS, l'implémentation de l'interface Destination est une superclasse abstraite de Queue et Topic, et une instance de Destination est un objet Queue ou Topic. Les interfaces indépendantes du domaine traitent une file d'attente ou une rubrique en tant que destination. Le domaine de messagerie pour un objet MessageProducer ou MessageConsumer varie selon que la destination est une file d'attente ou une rubrique.

Ainsi, dans IBM MQ classes for JMS, les objets des types suivants peuvent être des objets gérés :

- ConnectionFactory
- QueueConnectionFactory
- TopicConnectionFactory
- File d'attente
- Topic
- XAConnectionFactory
- XAQueueConnectionFactory
- XATopicConnectionFactory

## Architecture IBM MQ classes for JMS

IBM MQ classes for JMS possède une architecture en couches. La couche de code supérieure est une couche commune que tout fournisseur IBM JMS peut utiliser.

IBM MQ classes for JMS possède une architecture en couches comme illustré dans la [Figure 55](#), à la [page 166](#). La couche de code la plus haute est une couche commune qui peut être utilisée par n'importe quel fournisseur IBM JMS . Lorsqu'une application appelle une méthode JMS, tout traitement de l'appel qui n'est pas spécifique à un système de messagerie est effectué par la couche commune, qui fournit également une réponse cohérente à l'appel. Les traitements de l'appel spécifiques à un système de messagerie sont délégués à une couche inférieure. Dans le diagramme suivant, le fournisseur de messagerie IBM MQ apparaît dans la couche inférieure avec deux autres fournisseurs de messagerie (fournisseur de messagerie A et fournisseur de messagerie B.)

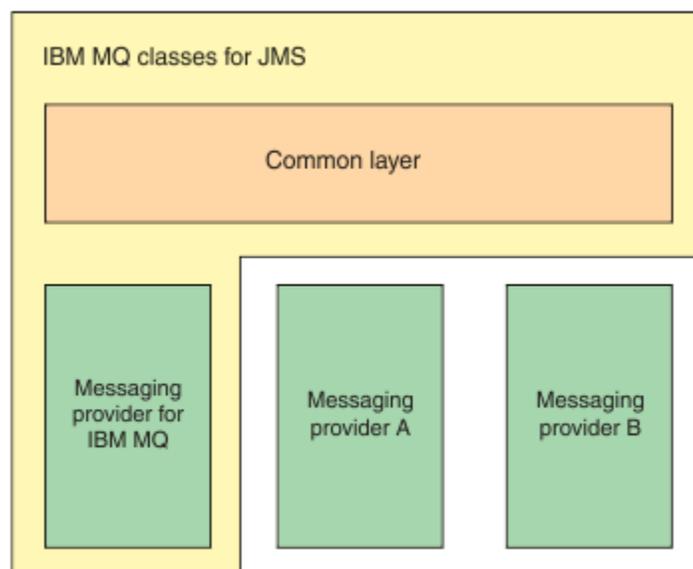


Figure 55. Architecture en couches pour les fournisseurs IBM JMS

Une architecture en couches répond aux objectifs suivants :

- Pour améliorer la cohérence du comportement des différents fournisseurs IBM JMS
- Faciliter l'écriture d'une application de passerelle entre deux systèmes de messagerie IBM
- Faciliter le portage d'une application d'un fournisseur IBM JMS vers un autre

## Prise en charge des objets gérés

IBM MQ classes for JMS prend en charge l'utilisation d'objets gérés.

Le flux logique dans une application JMS commence par des objets `ConnectionFactory` et `Destination`. L'application utilise un objet `ConnectionFactory` pour créer un objet `Connection`, qui représente la connexion active depuis l'application vers un serveur de messagerie. L'application utilise l'objet `Connection` pour créer un objet `Session`, qui est un contexte avec une seule unité d'exécution permettant de générer et de consommer des messages. L'application peut ensuite utiliser l'objet `Session` et l'objet `Destination` pour créer un objet `MessageProducer`, que l'application utilise pour envoyer des messages à la destination indiquée. La destination est soit une file d'attente, soit une rubrique du système de messagerie et est encapsulée par l'objet `Destination`. L'application peut aussi utiliser l'objet `Session` et un objet `Destination` pour créer un objet `MessageConsumer`, qu'elle utilise pour recevoir des messages envoyés à la destination indiquée.

La spécification JMS s'attend à ce que les objets `ConnectionFactory` et `Destination` soient des objets gérés. Un administrateur crée et gère des objets administrés dans un référentiel central, et une application JMS extrait ces objets à l'aide de Java Naming Directory Interface (JNDI). Le référentiel d'objets administrés peut aller d'un fichier simple à un répertoire LDAP (Lightweight Directory Access Protocol).

IBM MQ classes for JMS prend en charge l'utilisation d'objets gérés. Une application peut utiliser toutes les fonctions d'IBM MQ classes for JMS exposées via IBM MQ sans qu'aucune information spécifique à IBM MQ ne soit codée en dur dans l'application elle-même. Cela fournit à l'application une certaine indépendance par rapport à la configuration sous-jacente d'IBM MQ. Pour parvenir à cette indépendance, l'application peut utiliser JNDI pour extraire les fabriques de connexions et les destinations stockées sous forme d'objets gérés et utiliser uniquement les interfaces définies dans le package `javax.jms` pour effectuer des opérations de messagerie. Un administrateur peut utiliser l'outil d'administration IBM MQ JMS ou IBM MQ Explorer pour créer des objets administrés dans un référentiel central. Toutefois, un serveur d'applications fournit généralement son propre référentiel pour les objets gérés et ses propres

outils pour la création et la maintenance des objets. Une application Java EE peut donc utiliser JNDI pour extraire les objets gérés du référentiel du serveur d'applications ou d'un référentiel central.

### **Information associée**

[Configuration des ressources JMS](#)

## **Types de communications pris en charge sur les plateformes Java EE**

Sur la plateforme Java EE, les IBM MQ classes for JMS prennent en charge deux types de communication entre un composant d'une application et un gestionnaire de files d'attente IBM MQ.

Les deux types de communication suivants entre un composant d'une application et un gestionnaire de files d'attente IBM MQ sont pris en charge :

- Communication sortante
- Communication entrante

### **Communication sortante**

En utilisant directement l'API JMS, un composant d'application crée une connexion à un gestionnaire de files d'attente, puis envoie et reçoit ensuite des messages.

Par exemple, le composant d'application peut être un client d'application, un servlet, une page JavaServer Page (JSP), un bean entreprise Java (EJB), ou un bean géré par message (MDB). Dans ce type de communication, le conteneur du serveur d'applications fournit uniquement des fonctions de bas niveau pour le support des opérations de messagerie, comme le regroupement de connexions et la gestion des unités d'exécution.

### **Communication entrante**

Dans le cas de la communication entrante, un message parvenant à une destination est livré à un bean géré par message qui traite le message.

Les applications Java EE utilisent des beans gérés par message pour traiter les messages de manière asynchrone. Un bean géré par message agit en tant que programme d'écoute de message JMS et il est implémenté par une méthode `onMessage()` qui définit comment un message est traité. Un bean géré par message est déployé dans le conteneur d'EJB d'un serveur d'applications. La manière selon laquelle un bean géré par message est configuré varie en fonction du serveur d'applications utilisé, mais les informations de configuration doivent spécifier à quel gestionnaire de files d'attente se connecter, comment se connecter à ce dernier, quelle destination surveiller pour les messages et le comportement transactionnel du bean géré par message. Ces informations sont ensuite utilisées par le conteneur d'EJB. Lorsqu'un message remplissant les critères de sélection du bean géré par message parvient à la destination, indiquée, le conteneur d'EJB utilise IBM MQ classes for JMS pour extraire le message du gestionnaire de files d'attente, puis distribue le message au bean géré par message en appelant sa méthode `onMessage()`.

## **Relation avec IBM MQ classes for Java**

A partir d'IBM WebSphere MQ 7.0, IBM WebSphere MQ classes for JMS et IBM WebSphere MQ classes for Java sont implémentés en tant qu'homologues. Cette implémentation est différente des versions antérieures dans lesquelles l'implémentation d'IBM WebSphere MQ classes for JMS dépendait d'IBM WebSphere MQ classes for Java.

Pour les antérieures à IBM WebSphere MQ 7.0, IBM WebSphere MQ classes for JMS a été implémenté presque entièrement sous forme d'une couche de code par-dessus IBM WebSphere MQ classes for Java. Cet arrangement a entraîné une certaine confusion parmi les développeurs d'applications car la définition des zones ou l'appel des méthodes dans la classe `MQEnvironment` peut avoir des effets indésirables et inattendus sur le comportement d'exécution du code qui est écrit à l'aide d'IBM MQ classes for JMS. En outre, l'implémentation d'IBM MQ classes for JMS imposait des contraintes dans des domaines où l'API JMS ne se place pas naturellement sur IBM MQ classes for Java, et ces contraintes ont généré des problèmes concernant les performances d'exécution.

Depuis IBM WebSphere MQ 7.0, l'implémentation d'IBM MQ classes for JMS ne dépend plus d'IBM MQ classes for Java. IBM MQ classes for Java et IBM MQ classes for JMS sont des homologues qui utilisent une interface Java commune avec MQI. Cet arrangement augmente la portée pour l'optimisation des performances et signifie que la définition des zones ou l'appel des méthodes dans la classe MQEnvironment n'a aucun effet sur le comportement d'exécution du code qui est écrit à l'aide d'IBM MQ classes for JMS. La Figure 56, à la page 168 présente la relation entre IBM MQ classes for JMS et IBM MQ classes for Java dans les versions précédentes d'IBM WebSphere MQ classes for JMS et dans les versions antérieures à IBM WebSphere MQ 7.0 et comment cette relation a changé pour les versions ultérieures.

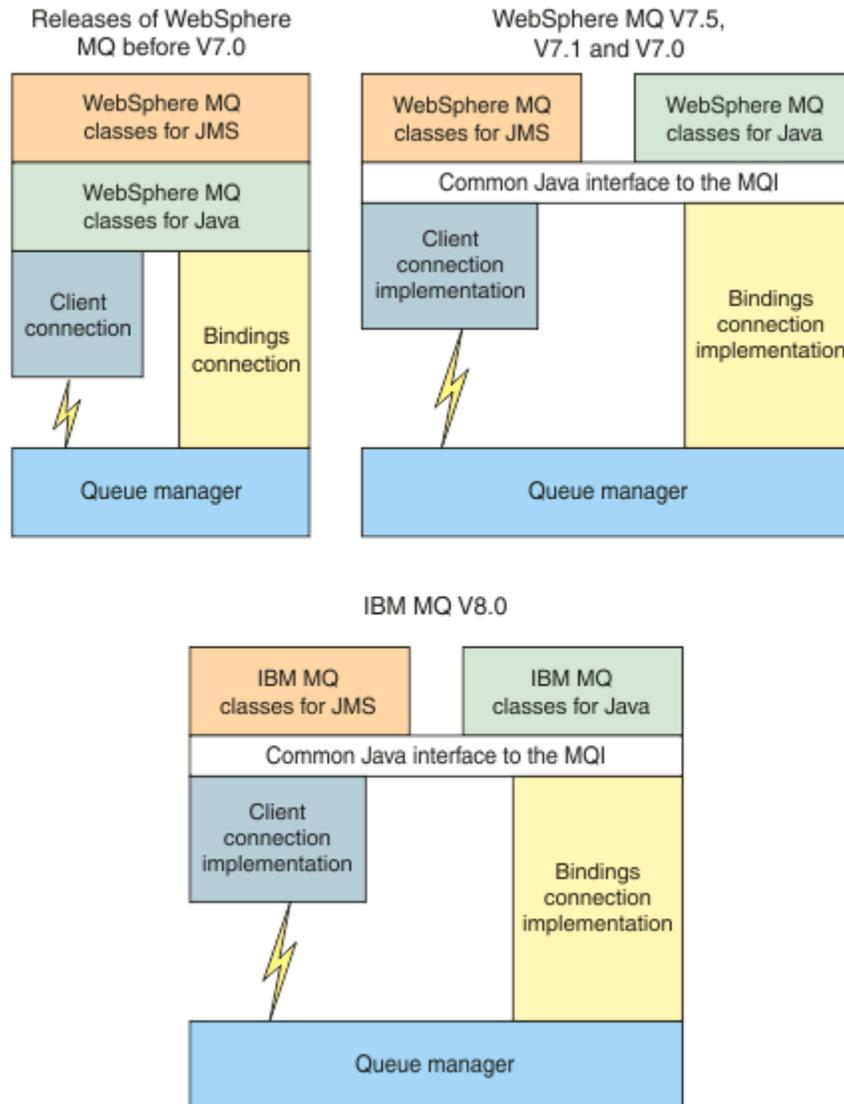


Figure 56. Relation entre IBM MQ classes for JMS et IBM MQ classes for Java

Pour assurer la compatibilité avec les versions antérieures à IBM WebSphere MQ 7.0, les classes d'exit de canal qui sont écrites dans Java peuvent encore utiliser les interfaces des IBM MQ classes for Java même si les classes d'exit de canal sont appelées à partir d'IBM MQ classes for JMS. Toutefois, l'utilisation des interfaces IBM MQ classes for Java signifie que vos applications dépendent toujours du fichier JAR IBM MQ classes for Java, `com.ibm.mq.jar`. Si vous ne souhaitez pas que `com.ibm.mq.jar` figure dans votre chemin d'accès aux classes, vous pouvez utiliser l'ensemble d'interfaces dans le package `com.ibm.mq.exits` à la place.

A partir d'IBM WebSphere MQ 7.0, vous pouvez créer et configurer des objets gérés JMS à l'aide d'IBM MQ Explorer.

## Fournisseur de messagerie IBM MQ

Le fournisseur de messagerie IBM MQ possède trois modes de fonctionnement : le mode normal, le mode normal avec restrictions et le mode de migration.

Le fournisseur de messagerie IBM MQ possède trois modes de fonctionnement :

- Le mode normal du fournisseur de messagerie IBM MQ
- Le mode normal avec restrictions du fournisseur de messagerie IBM MQ
- Le mode de migration du fournisseur de messagerie IBM MQ

Le mode de fournisseur de messagerie IBM MQ normal utilise toutes les fonctions d'un gestionnaire de files d'attente IBM MQ pour implémenter JMS. Ce mode est optimisé pour utiliser l'API JMS 2.0 et ses fonctionnalités.

Le mode normal du fournisseur de messagerie IBM MQ avec des restrictions utilise l'API JMS 2.0, mais pas les nouvelles fonctions IBM MQ 8.0, telles que les abonnements partagés, la distribution différée ou l'envoi asynchrone.

Le mode de migration du fournisseur de messagerie IBM MQ est basé sur la fonction IBM WebSphere MQ 6.0 et il utilise uniquement les fonctions qui étaient disponibles dans le gestionnaire de files d'attente IBM WebSphere MQ 6.0 pour implémenter JMS. Vous pouvez connecter à IBM WebSphere MQ 7.0, ou version ultérieure, des gestionnaires de files d'attente qui utilisent le mode de migration, mais vous ne pouvez pas utiliser les optimisations de la IBM WebSphere MQ 7.0. Ce mode autorise les connexions à l'une des versions de gestionnaire de files d'attente suivantes :

1. Gestionnaire de files d'attente IBM WebSphere MQ 7.0 ou version ultérieure en mode liaisons ou client, mais ce mode utilise uniquement les fonctions qui étaient disponibles pour un gestionnaire de files d'attente IBM WebSphere MQ 6.0.
2. Gestionnaire de files d'attente IBM WebSphere MQ 6.0 ou version antérieure en mode client.

Si vous voulez vous connecter à IBM Integration Bus à l'aide d'IBM MQ Enterprise Transport, utilisez le mode de migration. Si vous utilisez IBM MQ Real-Time Transport, le mode de migration est automatiquement sélectionné car vous avez sélectionné explicitement des propriétés dans l'objet de fabrication de connexions. La connexion à IBM Integration Bus avec IBM MQ Enterprise Transport suit les règles générales de sélection du mode qui sont décrites dans [Configuration de la propriété JMS PROVIDERVERSION](#).

### Tâches associées

[Configuration des ressources JMS](#)

## z/OS Concepts d'IBM MQ for z/OS

Certains concepts utilisés par IBM MQ for z/OS sont propres à la plateforme z/OS. Par exemple, le mécanisme de journalisation, les techniques de gestion de l'archivage, la disposition de l'unité de récupération et les groupes de partage de files d'attente sont fournis uniquement avec IBM MQ for z/OS. Pour plus d'informations sur ces concepts, consultez cette rubrique.

### Concepts associés

[«Le gestionnaire de files d'attente sur z/OS», à la page 170](#)

Avant de permettre à vos programmes d'application d'utiliser IBM MQ sur le système z/OS, vous devez installer le produit IBM MQ for z/OS et un gestionnaire de files d'attente. Le gestionnaire possède et gère les ensembles de ressources utilisés par IBM MQ.

[«L'initiateur de canal sous z/OS», à la page 171](#)

L'initiateur de canal fournit et gère les ressources qui activent la mise en file d'attente répartie IBM MQ. IBM MQ utilise des *agents MCA (Message Channel Agents)* pour envoyer des messages d'un gestionnaire de files d'attente à un autre.

[«Termes et tâches pour la gestion d'IBM MQ for z/OS», à la page 173](#)

Cette rubrique présente la terminologie et les tâches propres à IBM MQ for z/OS.

[«Files d'attente partagées et groupes de partage de files d'attente», à la page 176](#)

Vous avez la possibilité d'utiliser des files d'attentes partagées et des groupes de partage de files d'attentes pour implémenter la haute disponibilité des ressources IBM MQ. Les files d'attente partagées et les groupes de partage de files d'attente sont des fonctions propres à IBM MQ for z/OS sur la plateforme z/OS.

[«Mise en file d'attente intragroupe», à la page 227](#)

Cette section décrit la mise en files d'attente intra-groupe, une fonction IBM MQ for z/OS propre à la plateforme z/OS. Cette fonction n'est disponible que pour les gestionnaires de files d'attente définis dans un groupe de partage de files d'attente.

[«Gestion du stockage sur z/OS», à la page 242](#)

IBM MQ for z/OS requiert des structures de données permanentes et temporaires et utilise des ensembles de pages et des tampons mémoire pour archiver ces données. Ces rubriques contiennent d'autres informations sur le mode d'utilisation de ces ensembles de pages et de ces tampons par IBM MQ.

[«Journalisation dans IBM MQ for z/OS», à la page 246](#)

IBM MQ gère des *journaux* des modifications des données et des événements importants. Ces journaux peuvent, si nécessaire, servir à restaurer l'état précédent de certaines données.

[«Reprise et redémarrage sur z/OS», à la page 270](#)

Utilisez les liens contenus dans cette rubrique pour en savoir plus sur les fonctions de redémarrage et de reprise d'IBM MQ for z/OS.

[«Concepts de sécurité dans IBM MQ for z/OS», à la page 289](#)

Cette rubrique explique l'importance de la sécurité pour IBM MQ et les conséquences liées à l'absence de paramètres de sécurité adéquats sur votre système.

[«Disponibilité sur z/OS», à la page 296](#)

IBM MQ for z/OS propose de nombreuses fonctions pour la haute disponibilité. Cette rubrique décrit certaines des considérations en matière de disponibilité.

[«Disposition d'une unité de récupération sur z/OS», à la page 301](#)

Certaines applications transactionnelles peuvent utiliser une disposition d'unité de récupération GROUPE plutôt que QMGR lorsqu'elles sont connectées à un gestionnaire de files d'attente dans un groupe de partage de files d'attente en spécifiant le nom de groupe de partage de files d'attente plutôt que le nom du gestionnaire de files d'attente lorsqu'elles se connectent. Cela assouplit et renforce la reprise de transaction en supprimant le besoin de se reconnecter au même gestionnaire de files d'attente dans le groupe de partage de files d'attente.

### Référence associée

[«Définition de votre système sur z/OS», à la page 259](#)

IBM MQ for z/OS utilise plusieurs définitions d'objet par défaut et fournit des exemples de JCL permettant de créer ces objets. Cette rubrique permet de comprendre les objets par défaut et l'exemple de JCL.

[«Surveillance et statistiques sous IBM MQ for z/OS», à la page 300](#)

IBM MQ for z/OS est doté d'un ensemble de fonctions permettant de contrôler les gestionnaires de files d'attente et de collecter des statistiques.

z/OS

## Le gestionnaire de files d'attente sur z/OS

Avant de permettre à vos programmes d'application d'utiliser IBM MQ sur le système z/OS, vous devez installer le produit IBM MQ for z/OS et un gestionnaire de files d'attente. Le gestionnaire possède et gère les ensembles de ressources utilisés par IBM MQ.

### Gestionnaire de files d'attente

Un *gestionnaire de files d'attente* est un programme qui fournit des services de messagerie aux applications. Lorsque ces dernières utilisent l'interface MQI (Message Queue Interface), elles peuvent

effectuer des opérations d'insertion et d'extraction de messages sur les files d'attente. Le gestionnaire de files d'attente garantit la transmission des messages vers la file d'attente appropriée ou leur réacheminement vers un autre gestionnaire de files d'attente. Il traite les appels MQI qui lui sont adressés et les commandes qui lui sont soumises (quelle qu'en soit la source). Il génère les codes de fin appropriés pour chaque appel ou commande.

Les ressources gérées par le gestionnaire de files d'attente sont les suivantes :

- Les ensembles de pages contenant les définitions d'objet et les données de message IBM MQ
- Les journaux utilisés pour récupérer les messages et les objets en cas d'incident lié au gestionnaire de files d'attente
- La mémoire principale
- Les connexions via lesquelles différents environnements d'application (CICS, IMS et Batch) peuvent accéder à l'API IBM MQ
- L'initiateur de canal IBM MQ qui permet la communication entre IBM MQ sur le système z/OS et les autres systèmes

Le gestionnaire de files d'attente porte un nom à l'aide duquel les applications peuvent s'y connecter.

La [Figure 57](#), à la page 171 illustre un gestionnaire de files d'attente et présente les connexions à différents environnements d'application, ainsi que l'initiateur de canal.

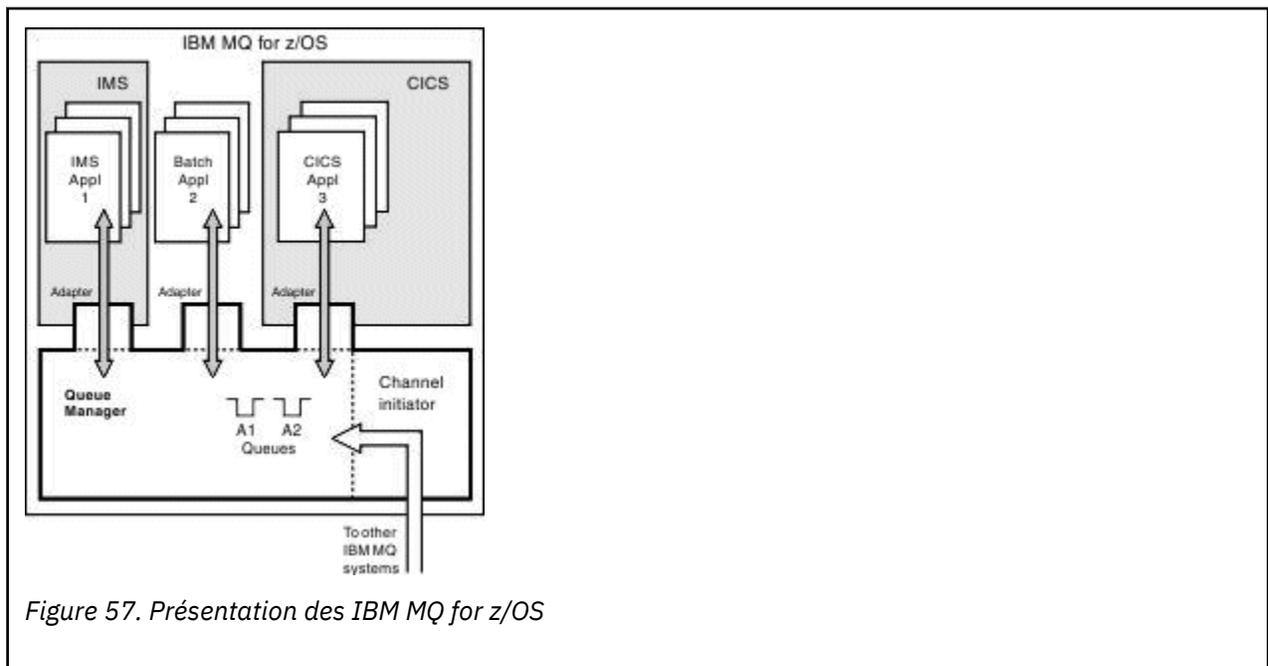


Figure 57. Présentation des IBM MQ for z/OS

## Le sous-système de gestion de files d'attente sur z/OS

Sur z/OS, IBM MQ s'exécute comme sous-système z/OS démarré lors du démarrage du système. Dans le sous-système, le gestionnaire de files d'attente démarre via une procédure JCL qui définit les jeux de données z/OS contenant des informations sur les journaux, ainsi que les définitions d'objet et les données de message (les jeux de données). Le sous-système et le gestionnaire de files d'attente portent le même nom, composé de quatre caractères au plus. Tous les noms des gestionnaires de files d'attente de votre réseau doivent être uniques, même s'ils se trouvent sur différents systèmes, sysplex ou plateformes.

### **z/OS** L'initiateur de canal sous z/OS

L'initiateur de canal fournit et gère les ressources qui activent la mise en file d'attente répartie IBM MQ. IBM MQ utilise des *agents MCA (Message Channel Agents)* pour envoyer des messages d'un gestionnaire de files d'attente à un autre.

Pour envoyer des messages du gestionnaire de files d'attente A au gestionnaire de files d'attente B, un MCA *sending* sur le gestionnaire de files d'attente A doit configurer un lien de communication vers le gestionnaire de files d'attente B. Un MCA *receiving* doit être démarré sur le gestionnaire de files d'attente B pour recevoir les messages du lien de communication. Ce chemin unidirectionnel composé de l'agent MCA émetteur, de la liaison et de l'agent MCA récepteur est appelé *canal*. L'agent MCA émetteur reçoit les messages d'une file d'attente de transmission et les envoie via un canal à l'agent MCA récepteur. Celui-ci reçoit les messages et les place dans les files d'attente de destination.

Dans IBM MQ for z/OS, les agents MCA expéditeurs et destinataires s'exécutent tous dans l'initiateur de canal (l'initiateur de canal est également appelé *déplaceur*). L'initiateur de canal s'exécute en tant qu'espace adresse z/OS sous le contrôle du gestionnaire de files d'attente. Un seul initiateur de canal peut être connecté à un gestionnaire de files d'attente. Cet initiateur s'exécute au sein de la même image z/OS que le gestionnaire de files d'attente. Plusieurs milliers de processus d'agents MCA peuvent s'exécuter simultanément dans l'initiateur de canal.

La Figure 58, à la page 172 présente deux gestionnaires de files d'attente dans un sysplex. Chaque gestionnaire est doté d'un initiateur de canal et d'une file d'attente locale. Les messages envoyés par les gestionnaires de files d'attente sur AIX et Windows sont placés dans la file d'attente locale, d'où ils sont récupérés par une application. Les messages de réponse sont renvoyés par la même voie.

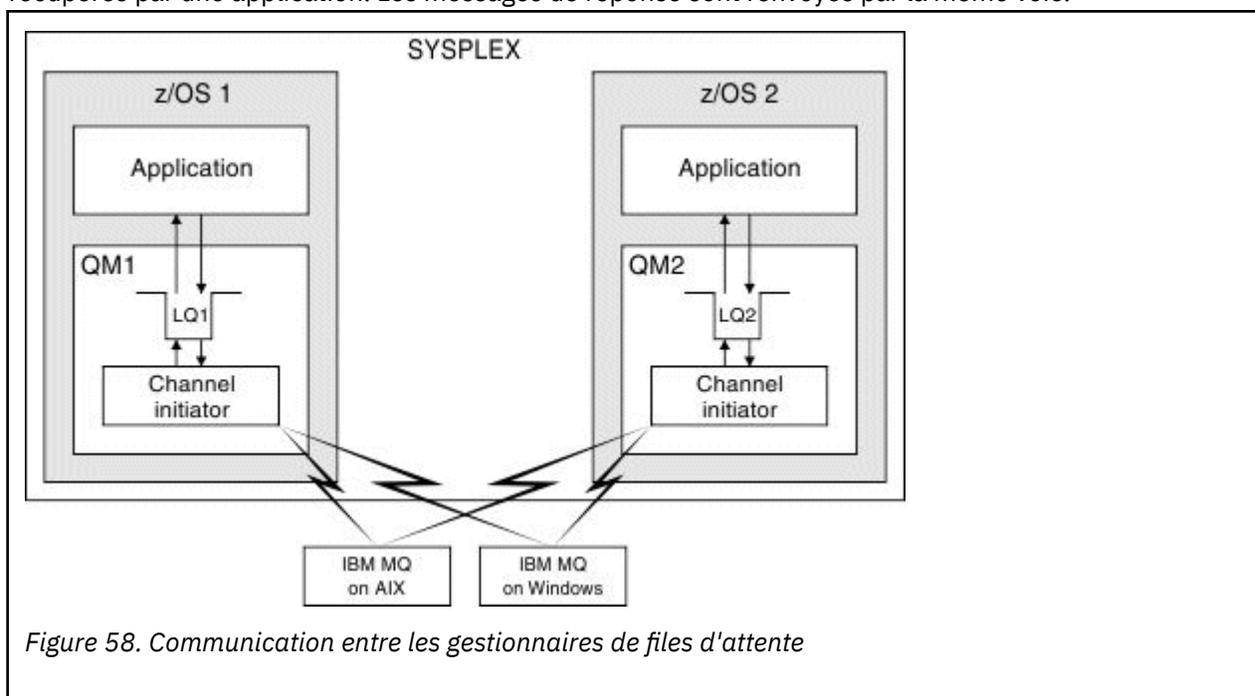


Figure 58. Communication entre les gestionnaires de files d'attente

L'initiateur de canal contient également d'autres processus liés à la gestion des canaux. Ces processus sont les suivants :

#### Programmes d'écoute

Ces processus écoutent les demandes de canal de communications entrantes sur un sous-système de communication tel que TCP et démarrent un agent MCA nommé lorsqu'une demande entrante est reçue.

#### Superviseur

Le superviseur gère l'espace adresse de l'initiateur de canal. Il est par exemple responsable du redémarrage des canaux après un échec.

#### Serveurs de noms

Le serveur de noms est utilisé pour résoudre les noms TCP en adresses.

#### Tâches TLS

Ces tâches sont utilisées pour les opérations de chiffrement et de déchiffrement et pour vérifier les listes de révocation de certificat.

## Enregistrements SMF de l'initiateur de canal

L'initiateur de canal (CHINIT) peut produire des enregistrements de statistiques SMF et des enregistrements de comptabilité comportant des informations sur les tâches et les canaux.

L'initiateur de canal CHINIT peut produire des enregistrements de statistiques SMF et des enregistrements de comptabilité comportant ces différents types d'informations :

- Les tâches : répartiteur, adaptateur, serveur de noms de domaine (DNS) et SSL. Ces tâches constituent ce qu'on appelle des statistiques CHINIT.
- Canaux : fournit des informations comptables similaires à celles disponibles avec la commande DIS CHSTATUS. Cela est appelé comptabilité de canal.

IBM MQ for Multiplatforms fournit des informations similaires en écrivant des messages PCF à SYSTEM.ADMIN.STATISTICS.QUEUE. Voir [Données des messages de statistiques de canal](#) pour plus d'informations sur l'enregistrement des statistiques sur IBM MQ for Multiplatforms.

### Données statistiques

Vous pouvez utiliser ces données pour rechercher les informations suivantes :

- Déterminer si des tâches CHINIT supplémentaires sont requises, comme le nombre de tâches TCB SSL et la quantité d'UC utilisée par ces tâches.
- Le temps moyen des demandes sur ces tâches.
- La demande d'une durée la plus longue dans l'intervalle et le moment de la journée où elle s'est produite, pour les tâches DNS et SSL. Vous pouvez corréliser ce moment de la journée aux problèmes susceptibles de survenir avec ce canal.

### Données de compte

Vous pouvez utiliser ces données pour surveiller l'utilisation des canaux et rechercher les informations suivantes :

- Les canaux ayant le débit le plus élevé.
- Le taux d'envoi des messages et le taux d'envoi des données en Mo/seconde.
- La taille de lots obtenue. Si la taille de lots obtenue est proche de celle indiquée pour le canal, il est possible que le canal atteigne sa limite d'envoi des messages.

Vous utilisez les commandes [START TRACE](#) et [STOP TRACE](#) pour contrôler la collecte de la trace comptable et de la trace de statistique. Vous pouvez utiliser les options [STATCHL](#) et [STATACLS](#) sur le canal et le gestionnaire de files d'attente pour contrôler si les canaux génèrent des données SMF.

## Termes et tâches pour la gestion d'IBM MQ for z/OS

Cette rubrique présente la terminologie et les tâches propres à IBM MQ for z/OS.

Certains des termes et tâches requis dans le cadre de la gestion d'IBM MQ for z/OS sont propres à la plateforme z/OS. La liste suivante présente certains de ces termes et tâches.

- [Files d'attente partagées](#)
- [Ensembles de pages et pools de mémoire tampon](#)
- [Journalisation](#)
- [Personnalisation de l'environnement du gestionnaire de files d'attente](#)
- [Redémarrage et reprise](#)
- [Sécurité](#)
- [Disponibilité](#)
- [Manipulation d'objets](#)

- [Surveillance et statistiques](#)
- [Environnements d'application](#)

## Files d'attente partagées

Les files d'attente peuvent être *non partagées* (elles appartiennent à un seul gestionnaire de files d'attente et ne sont accessibles que par celui-ci) ou *partagées* (elles appartiennent à un *groupe de partage de files d'attente*). Un groupe de partage de files d'attente est constitué d'un certain nombre de gestionnaires de files d'attente s'exécutant dans un même sysplex z/OS pouvant accéder simultanément aux mêmes définitions d'objet IBM MQ et aux mêmes données de message. Dans un groupe de partage de files d'attente, les définitions d'objet partageables sont stockées dans une base de données Db2. Les messages des files d'attente partagées sont conservés dans une ou plusieurs structures d'unité de couplage. Si les données de message sont trop volumineuses pour être stockées directement dans la structure (d'une taille supérieure à 63 ko) ou que le message est suffisamment volumineux pour être sélectionné par les règles définies lors de l'installation en vue de son déchargement, les informations de contrôle du message sont toujours stockées dans l'entrée de l'unité de couplage mais les données de message sont déchargées dans un fichier de messages partagés ou dans une base de données Db2. Les fichiers de messages partagés, la base de données Db2 et les structures d'unité de couplage sont des ressources gérées conjointement par tous les gestionnaires de files d'attente du groupe.

## Ensembles de pages et pools de mémoire tampon

Lorsqu'un message est placé dans une file d'attente non partagée, le gestionnaire de files d'attente stocke ces données dans un ensemble de pages de sorte à pouvoir les récupérer lorsqu'une opération ultérieure extrait un message de la même file d'attente. Si ce message est supprimé de la file d'attente, l'espace de l'ensemble de pages contenant ces données est ensuite libéré pour être réutilisé. Au fur et à mesure que le nombre de messages contenus dans la file d'attente augmente, l'espace utilisé dans l'ensemble de pages augmente et au fur et à mesure que le nombre de messages diminue, l'espace utilisé diminue.

Pour réduire les coûts de performances liés à l'écriture et à la lecture des données dans les ensembles de pages, le gestionnaire de files d'attente place les mises à jour dans la mémoire tampon du processeur. La quantité de mémoire utilisée pour mettre en mémoire tampon l'accès à l'ensemble de pages est contrôlé par des objets IBM MQ appelés *pools de mémoire tampon*.

Pour plus d'informations sur les ensembles de pages et les pools de mémoire tampon, voir [Gestion de l'archivage](#).

## Journalisation

Les modifications apportées aux objets contenus dans les ensembles de pages et les opérations exécutées sur les messages persistants sont enregistrées en tant qu'enregistrement de journal. Ces enregistrements sont écrits dans un jeu de données de journal appelé *journal actif*. Le nom et la taille du jeu de données de journal actif sont contenus dans un fichier appelé *fichier d'amorce*.

Lorsque le jeu de données de journal actif se remplit, le gestionnaire de files d'attente passe à un autre fichier pour pouvoir poursuivre la journalisation et copie le contenu du jeu de données de journal actif dans un fichier *d'archivage*. Les informations relatives à ces actions, notamment le nom du fichier d'archivage, sont contenues dans le fichier d'amorce. Sur le plan conceptuel, les jeux de données de journal actifs sont disposés en anneau que le gestionnaire de files d'attente parcourt. Lorsqu'un journal actif est rempli, ses données sont déchargées dans un fichier d'archivage et le fichier actif peut être réutilisé.

Pour plus d'informations sur le journal et les fichiers d'amorce, voir [«Journalisation dans IBM MQ for z/OS»](#), à la page 246.

## Personnalisation de l'environnement du gestionnaire de files d'attente

Lorsque le gestionnaire de files d'attente démarre, un ensemble de paramètres d'initialisation contrôlant le fonctionnement du gestionnaire de files d'attente est lu. En outre, les fichiers contenant les commandes IBM MQ sont lus, et les commandes qui s'y trouvent sont exécutées. En général, ces fichiers contiennent les définitions des objets système requis pour l'exécution d'IBM MQ, et vous pouvez personnaliser ces définitions afin de définir ou initialiser les objets IBM MQ nécessaires à votre environnement d'exploitation. Une fois ces fichiers lus, les objets qu'ils définissent sont stockés dans un ensemble de pages ou dans Db2.

Pour plus d'informations sur les paramètres d'initialisation et les objets système, voir [«Définition de votre système sur z/OS»](#), à la page 259.

## Reprise et redémarrage

A tout moment au cours de l'exécution d'IBM MQ, la mémoire processeur peut contenir des modifications n'ayant pas encore été écrites dans l'ensemble de pages. Ces modifications sont écrites dans l'ensemble de pages utilisée la moins récemment par une tâche d'arrière-plan du gestionnaire de files d'attente.

Si le gestionnaire de files d'attente s'arrête de façon anormale, la phase de reprise du gestionnaire de files d'attente peut récupérer les modifications perdues car les données des messages persistants sont contenues dans les enregistrements de journal. Cela signifie qu'IBM MQ peut récupérer les modifications apportées aux objets et aux données des messages persistants jusqu'au point de défaillance.

Si une erreur liée à l'unité de couplage survient sur un gestionnaire de files d'attente membre d'un groupe de partage, les messages persistants de cette file d'attente peuvent uniquement être récupérés si vous avez sauvegardé la structure de votre unité de couplage.

Pour plus d'informations sur la reprise et le redémarrage, voir [«Reprise et redémarrage sur z/OS»](#), à la page 270.

## Sécurité

Vous pouvez utiliser un gestionnaire de sécurité externe tel que Security Server (appelé auparavant RACF) pour protéger les ressources dont IBM MQ est propriétaire et qu'il gère à partir de l'accès par des utilisateurs non autorisés. Vous pouvez également utiliser le protocole TLS (Transport Layer Security) pour la sécurité des canaux. TLS fait partie intégrante du produit IBM MQ.

Pour plus d'informations sur la sécurité IBM MQ, voir [«Concepts de sécurité dans IBM MQ for z/OS»](#), à la page 289.

## Disponibilité

Plusieurs fonctions d'IBM MQ sont conçues pour améliorer la disponibilité du système dans le cas d'une défaillance du gestionnaire de files d'attente ou du sous-système de communications. Pour plus d'informations sur ces fonctions, voir [«Disponibilité sur z/OS»](#), à la page 296.

## Manipulation d'objets

Lorsque le gestionnaire de files d'attente s'exécute, vous pouvez manipuler les objets IBM MQ via une interface de la console z/OS ou via un utilitaire d'administration utilisant les services ISPF sous TSO. Les deux mécanismes vous permettent de définir, modifier ou supprimer les objets IBM MQ. Vous pouvez également contrôler et afficher le statut des différentes fonctions du gestionnaire de files d'attente et d'IBM MQ.

Vous pouvez également manipuler les objets IBM MQ à l'aide d'IBM MQ Explorer, une interface utilisateur graphique proposant un mode visuel d'utilisation des files d'attente, des gestionnaires de files d'attente et des autres objets.

Pour plus d'informations sur ces fonctions, voir [Emission de commandes](#).

## Surveillance et statistiques

Plusieurs fonctions sont disponibles pour surveiller les gestionnaires de files d'attente et les initiateurs de canal. Vous pouvez également collecter des statistiques pour les tâches de comptabilité et d'évaluation des performances.

Pour plus d'informations sur ces fonctions, voir [«Surveillance et statistiques sous IBM MQ for z/OS»](#), à la page 300.

## Environnements d'application

Lorsque le gestionnaire de files d'attente est démarré, les applications peuvent s'y connecter et commencer à utiliser l'API IBM MQ. Il peut s'agir des applications CICS, IMS, Batch ou WebSphere Application Server. Les applications IBM MQ peuvent aussi accéder aux applications sous les systèmes CICS et IMS n'ayant pas détecté la présence d'IBM MQ, en utilisant des passerelles CICS et IMS.

Pour plus d'informations sur ces fonctions, voir [«IBM MQ et autres produits z/OS»](#), à la page 304.

Pour plus d'informations sur l'écriture des applications IBM MQ, voir la documentation suivante :

- [Développement d'applications](#)
- [Utilisation de C++](#)
- [Utilisation d'IBM MQ classes for Java](#)

z/OS

## Files d'attente partagées et groupes de partage de files d'attente

Vous avez la possibilité d'utiliser des files d'attentes partagées et des groupes de partage de files d'attentes pour implémenter la haute disponibilité des ressources IBM MQ. Les files d'attente partagées et les groupes de partage de files d'attente sont des fonctions propres à IBM MQ for z/OS sur la plateforme z/OS.

Cette section décrit les attributs et les avantages et fournit des informations sur la manière dont plusieurs gestionnaires de files d'attente peuvent partager les mêmes files d'attente et les mêmes messages sur ces files d'attente.

### Concepts associés

[«Qu'est-ce qu'une file d'attente partagée ?»](#), à la page 177

Une file d'attente partagée et un type de file d'attente local. Les messages présents sur cette file d'attente peuvent faire l'objet d'un accès par un ou plusieurs gestionnaires de files d'attente se trouvant sur un sysplex.

[«Qu'est-ce qu'un groupe de partage de files d'attente ?»](#), à la page 178

Un groupe de gestionnaires de files d'attente accédant aux mêmes files d'attente partagées est appelé groupe de partage de files d'attente. Chaque membre du groupe de partage de files d'attente a accès au même ensemble de files d'attente partagées.

[«Emplacement des messages de files d'attente partagées»](#), à la page 180

Chaque message dans une file d'attente partagée est représenté par une entrée dans une structure de liste de l'unité de couplage z/OS. Si les données de message sont trop volumineuses pour être stockées dans la même entrée, elles sont déchargées dans un fichier de messages partagés ou dans Db2.

[«Avantages liés à l'utilisation des files d'attente partagées»](#), à la page 197

Les files d'attente partagées permettent aux applications IBM MQ d'être évolutives, hautement disponibles et d'implémenter l'équilibrage de charge.

[«Mise en file d'attente répartie et groupes de partage de files d'attente»](#), à la page 220

La mise en file d'attente répartie et les groupes de partage de files d'attente sont les deux techniques que vous pouvez utiliser pour améliorer la disponibilité des systèmes d'application. Cette rubrique permet de rechercher des informations supplémentaires concernant ces techniques.

[«Influence de la distribution de charge de travail avec des files d'attente partagées»](#), à la page 225

Cette rubrique permet de comprendre comment certains facteurs liés aux files d'attente partagées ont une incidence sur la distribution de la charge de travail dans un groupe de partage de files d'attente.

### Référence associée

«Où trouver des informations supplémentaires sur les files d'attente partagées et les groupes de partage de files d'attente ?», à la page 226

Le tableau ci-dessous fournit des informations supplémentaires sur la façon dont IBM MQ for z/OS utilise les files d'attente partagées et les groupes de partage de files d'attente.

## Qu'est-ce qu'une file d'attente partagée ?

Une file d'attente partagée et un type de file d'attente local. Les messages présents sur cette file d'attente peuvent faire l'objet d'un accès par un ou plusieurs gestionnaires de files d'attente se trouvant sur un sysplex.

### Groupe de partage de files d'attente

Les gestionnaires de files d'attente qui accèdent au même ensemble de files d'attente partagée forment un groupe appelé *groupe de partage de files d'attente*.

### Tous les gestionnaires de files d'attente peuvent accéder aux messages

Chaque gestionnaire de files d'attente du groupe de partage de files d'attente peut accéder à une file d'attente partagée. Vous pouvez donc placer un message dans une file d'attente partagée se trouvant sur un gestionnaire de files d'attente et obtenir ce même message à partir d'un autre gestionnaire de files d'attente. Cette procédure constitue un mécanisme rapide de communication au sein d'un groupe de partage de files d'attente ne requérant aucun canal actif entre les gestionnaires de files d'attente.

IBM WebSphere MQ 7.1 et les versions ultérieures prennent en charge le déchargement des messages vers Db2 ou un fichier SMDS (shared message data set). Le déchargement des messages de toutes tailles peut être configuré.

Dans les versions précédentes d'IBM MQ, les messages de grande taille (> 63 Ko) contiennent une marque de réservation stockée dans l'unité de couplage (4 ko), et leurs données de message sont stockées dans Db2.

La [Figure 59](#), à la [page 178](#) présente trois gestionnaires de files d'attente et une unité de couplage formant un groupe de partage de files d'attente. Les trois gestionnaires peuvent accéder à la file d'attente partagée se trouvant dans l'unité de couplage.

Une application peut se connecter à l'un des gestionnaires de files d'attente du groupe de partage de files d'attente. Tous les gestionnaires de files d'attente du groupe de partage de files d'attente pouvant accéder aux files d'attente partagées, l'application ne dépend pas de la disponibilité d'un gestionnaire de files d'attente donnée. N'importe quel gestionnaire du groupe de partage de files d'attente peut gérer la file d'attente.

Une plus grande disponibilité est ainsi possible, car tous les autres gestionnaires de files d'attente du groupe de partage de files d'attente peuvent continuer à traiter la file d'attente si un problème lié à l'un des gestionnaires survient.

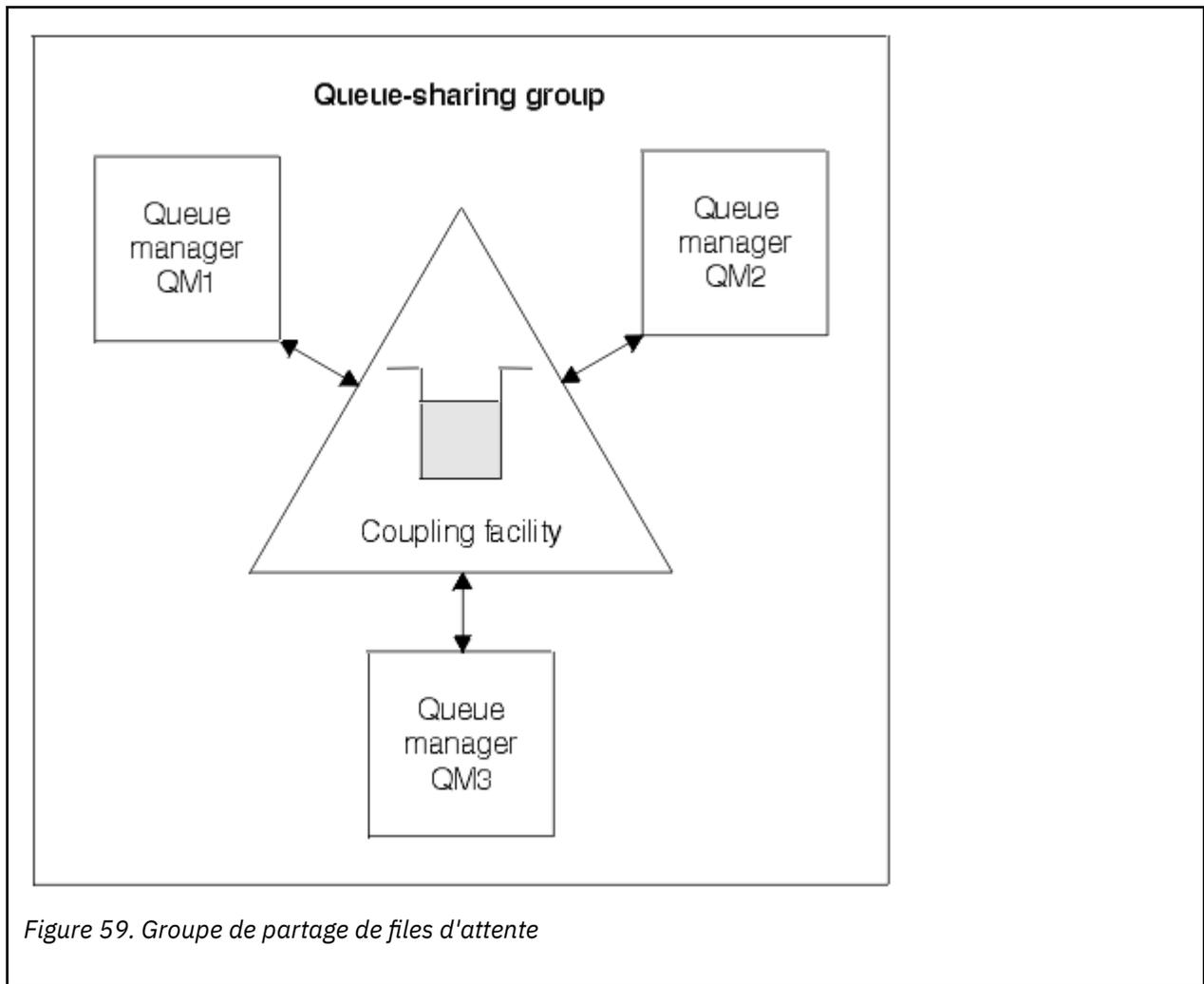


Figure 59. Groupe de partage de files d'attente

## La définition de file d'attente est partagée par tous les gestionnaires de files d'attente

Les définitions de files d'attente partagées sont stockées dans la table de base de données Db2 OBJ\_B\_QUEUE. Il vous suffit donc de définir la file d'attente une fois. Tous les gestionnaires de files d'attente du groupe de partage de files d'attente peuvent ensuite y accéder. Le nombre de définitions à créer est donc moindre.

En revanche, la définition d'une file d'attente non partagée est stockée dans l'ensemble de pages zéro du gestionnaire de files d'attente propriétaire de la file d'attente (voir la description dans [Ensembles de pages](#)).

Il est impossible de définir une file d'attente partagée si une file d'attente portant le même nom a déjà été définie dans les ensembles de pages du gestionnaire de files d'attente procédant à la définition. De même, il est impossible de définir une version locale d'une file d'attente dans les ensembles de pages du gestionnaire de files d'attente si une file d'attente partagée de même nom existe déjà.

### ► z/OS Qu'est-ce qu'un groupe de partage de files d'attente ?

Un groupe de gestionnaires de files d'attente accédant aux mêmes files d'attente partagées est appelé groupe de partage de files d'attente. Chaque membre du groupe de partage de files d'attente a accès au même ensemble de files d'attente partagées.

Le nom des groupes de partage de files d'attente comporte jusqu'à quatre caractères. Il doit être unique sur votre réseau et être différent de celui d'un gestionnaire de files d'attente.

La Figure 60, à la page 179 illustre un groupe de partage de files d'attente contenant deux gestionnaires de files d'attente. Chaque gestionnaire est doté d'un initiateur de canal et de ses propres ensembles de pages locaux et jeux de données de journal.

Chaque membre du groupe de partage de files d'attente doit également se connecter à un système Db2. Les systèmes Db2 doivent se trouver dans le même groupe de partage de données Db2 afin que les gestionnaires de files d'attente puissent accéder au référentiel partagé Db2 utilisé pour contenir les définitions d'objet partagé. Ces définitions peuvent concerner n'importe quel type d'objet IBM MQ (par exemple une file d'attente et un canal) défini une seule fois, puis pouvant être utilisé par n'importe quel gestionnaire de files d'attente du groupe. On les appelle des définitions *globales*. Elles sont décrites dans Définitions globales et privées.

Plusieurs groupes de partage de files d'attente peuvent faire référence à un groupe de partage de données particulier. Vous définissez le nom du sous-système Db2 et le groupe de partage de données utilisé par un gestionnaire de files d'attente dans les paramètres système IBM MQ lors du démarrage.

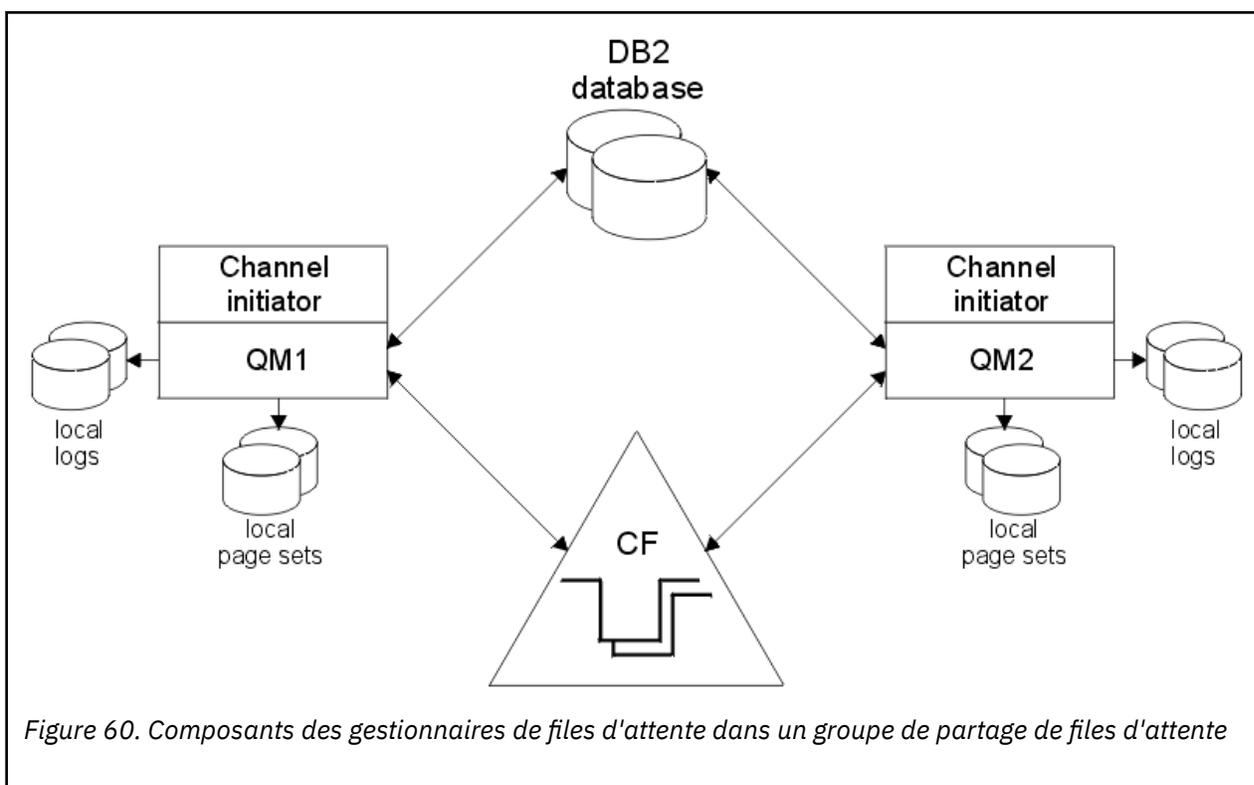


Figure 60. Composants des gestionnaires de files d'attente dans un groupe de partage de files d'attente

Une fois qu'un gestionnaire de files d'attente a rejoint un groupe de partage de files d'attente, il a accès aux objets partagés définis pour ce groupe et vous pouvez l'utiliser pour définir de nouveaux objets partagés dans ce groupe. Si des files d'attente partagées sont définies dans le groupe, vous pouvez utiliser ce gestionnaire pour y placer ou en extraire des messages. Chaque gestionnaire de files d'attente du groupe peut extraire les messages se trouvant dans une file d'attente partagée.

Vous pouvez entrer une commande MQSC une fois et l'exécuter sur tous les gestionnaires de files d'attente du groupe de partage comme si elle avait été entrée pour chaque gestionnaire séparément. L'attribut de *portée d'exécution* est utilisé à cet effet. Il est décrit à la section Direction des commandes vers des gestionnaires de files d'attente différents.

Lorsqu'un gestionnaire de files d'attente s'exécute en tant que membre d'un groupe de partage de files d'attente, il doit être possible de faire la distinction entre les objets IBM MQ définis de manière privée dans ce gestionnaire de files d'attente et les objets IBM MQ définis de manière globale et disponibles pour tous les gestionnaires de files d'attente de le groupe de partage de files d'attente. L'attribut de

*disposition de groupe de partage de files d'attente* est utilisé à cet effet. Il est décrit à la section [Définitions globales et privées](#).

Vous pouvez définir un ensemble unique de profils de sécurité contrôlant l'accès aux objets IBM MQ n'importe où dans le groupe. Le nombre de profils à définir est, par conséquent, considérablement réduit.

Un gestionnaire de files d'attente peut appartenir à un seul groupe de partage de files d'attente et tous les gestionnaires de files d'attente du groupe doivent se trouver dans le même sysplex. Vous définissez le groupe de partage de files d'attente auquel le gestionnaire de files d'attente appartient dans les paramètres système, au démarrage.

## **Emplacement des messages de files d'attente partagées**

Chaque message dans une file d'attente partagée est représenté par une entrée dans une structure de liste de l'unité de couplage z/OS. Si les données de message sont trop volumineuses pour être stockées dans la même entrée, elles sont déchargées dans un fichier de messages partagés ou dans Db2.

Si la structure d'unité de couplage a été configurée pour utiliser System Class Memory (SCM), IBM MQ peut l'utiliser sans configuration supplémentaire. Voir le chapitre 8 du manuel [IBM MQ V8 Features and Enhancements](#).

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

## **Stockage des messages de files d'attente partagées**

Les messages placés dans les files d'attente partagées ne sont pas stockés dans les ensembles de pages et n'utilisent pas les pools de mémoire tampon.

Les messages dans les files partagées ont des entrées dans des structures de liste de l'unité de couplage z/OS (CF). Plusieurs gestionnaires de files d'attente sur un même sysplex peuvent accéder à ces messages à l'aide de la structure de liste de l'unité de couplage.

Les données des messages des files d'attente partagées de petite taille sont normalement inclus dans l'entrée de l'unité de couplage. Pour les messages de plus grande taille, les données peuvent être contenues dans un fichier de messages partagés (SMDS) ou sous la forme d'un ou plusieurs objets binaires de grande taille (BLOB) dans une table Db2 partagée par un groupe de partage des données Db2. Les données de message dont la taille est supérieure à 63 ko sont toujours déchargées dans un fichier de messages partagés ou Db2. Les messages dont la taille est inférieure peuvent éventuellement aussi être déchargés de la même manière afin d'occuper moins d'espace dans la structure d'unité de couplage. Pour plus d'informations, voir [«Définition des options de déchargement pour les messages partagés»](#), à la page 182.

Les messages placés dans une file d'attente partagée sont référencés dans une structure d'unité de couplage jusqu'à ce qu'elles soient récupérées par un appel MQGET. Les opérations de l'unité de couplage permettent de :

- Rechercher le message pouvant être extrait suivant.
- Verrouiller les messages non validés sur les files d'attente partagées.
- Notifier les gestionnaires de files d'attente intéressés de l'arrivée des messages validés.

Les opérations MQPUT et MQGET exécutées sur les messages persistants sont enregistrées dans le journal du gestionnaire de files d'attente exécutant ces opérations. Cela permet de réduire le risque de perdre des données en raison d'une défaillance de l'unité de couplage.

## L'unité de couplage

Les messages contenus dans les files d'attente partagées sont référencés dans une unité de couplage. Cette unité se trouve en dehors des images z/OS présentes sur le sysplex et elle est en général configurée pour s'exécuter sur une machine différente. L'unité de couplage peut donc résister aux défaillances logicielles et vous pouvez la configurer pour résister aux défaillances matérielles et aux coupures de courant. Cela signifie que les messages stockés sur cette unité sont donc hautement disponibles.

Chaque structure de liste d'unité de couplage utilisée par IBM MQ est dédiée à un groupe de partage de files d'attente spécifique, mais une unité de couplage peut contenir des structures pour plusieurs groupes de partage de files d'attente. Les gestionnaires de files d'attente de différents groupes de partage ne peuvent pas partager de données. Le nombre maximal de gestionnaires de files d'attente d'un groupe de partage pouvant se connecter simultanément à une structure de liste de l'unité de couplage est égal à 32.

La même structure de liste d'une unité de couplage peut contenir 512 files d'attente partagées au maximum. La quantité totale de données de message stockées dans une structure est limitée par la capacité de celle-ci. Toutefois, avec **CFLEVEL (5)**, vous pouvez utiliser les paramètres de déchargement pour décharger les données des messages d'une taille inférieure à 63 ko. Vous pouvez alors stocker un plus grand nombre de messages dans la structure, bien qu'une entrée de l'unité de couplage et 512 octets de données au moins soient nécessaire pour chaque message.

La taille de la structure de liste est limitée par les facteurs suivants :

- Elle doit résider sur une seule unité de couplage.
- Elle peut partager la mémoire disponible de l'unité de couplage avec d'autres structures d'IBM MQ et d'autres produits.

Les structures de liste de l'unité de couplage peuvent être associées à une mémoire de classe d'archivage. Dans certaines situations, cette mémoire de classe d'archivage peut être utile en combinaison avec des files d'attente partagées. Pour plus d'informations, voir [«Utilisation de la mémoire de classe d'archivage avec des files d'attente partagées»](#), à la page 199.

**Remarque :** Lorsque vous utilisez IBM MQ, vous pouvez chiffrer une structure d'unité de couplage. Pour plus d'informations, voir [Chiffrement des données de structure d'unité de couplage](#).

## Planification de la taille de la structure de l'unité de couplage

Si vous souhaitez obtenir des instructions sur le dimensionnement des structures d'unité de couplage, vous pouvez utiliser le Supportpac MP16: [IBM MQ for z/OS Capacity planning and tuning](#). Vous pouvez également utiliser l'outil Web [CFSizer](#) fourni par IBM pour faciliter le dimensionnement de taille des unités de couplage.

## L'objet de structure de l'unité de couplage

L'utilisation par le gestionnaire de files d'attente d'une structure d'unité est définie dans un objet IBM MQ de structure d'unité de couplage (CFSTRUCT).

Ces objets de structure sont stockés dans Db2.

Lorsque vous utilisez des commandes ou des définitions z/OS associées à une structure d'unité de couplage, les quatre premiers caractères du nom du groupe de partage de files d'attente sont requis. Cependant, un objet IBM MQ CFSTRUCT existe toujours dans un groupe de partage de files d'attente, et son nom ne contient donc pas ces caractères. Par exemple, l'objet CFSTRUCT(MYDATA) défini dans le groupe de partage de files d'attente dont le nom commence par SQ03 utilise la structure de liste de l'unité de couplage SQ03MYDATA.

Les structures de l'unité de couplage sont associées à un attribut CFLEVEL qui détermine leur fonctionnalité :

- 1 ou 2 : niveau pouvant être utilisé pour les messages non persistants dont la taille est inférieure à 63 ko
- 3 : niveau pouvant être utilisé pour les messages persistants et non persistants dont la taille est inférieure à 63 ko
- 4 : niveau pouvant être utilisé pour les messages persistants et non persistants dont la taille peut atteindre 100 Mo
- 5 - niveau pouvant être utilisé pour les messages persistants et non persistants dont la taille peut atteindre 100 Mo et déchargés sélectivement vers des fichiers de messages partagés (SMDS) ou Db2.

## Sauvegarde et reprise de l'unité de couplage

Vous pouvez sauvegarder les structures de liste d'unité de couplage à l'aide de la commande IBM MQ `BACKUP CFSTRUCT`. Cette commande permet de placer une copie des messages persistants se trouvant dans la structure de l'unité de couplage dans le jeu de données de journal actif du gestionnaire de files d'attente en charge de la sauvegarde, et d'écrire un enregistrement de la sauvegarde dans Db2.

Si l'unité de couplage échoue, vous pouvez utiliser la commande IBM MQ `RECOVER CFSTRUCT`. Elle utilise l'enregistrement de sauvegarde de Db2 pour localiser et restaurer les messages persistants de la sauvegarde de la structure de l'unité de couplage. Toutes les opérations postérieures à la dernière sauvegarde sont réexécutées à l'aide des journaux de tous les gestionnaires de files d'attente du groupe de partage et l'état de la structure de l'unité de couplage au moment précédant l'échec est restauré.

Pour plus d'informations, voir les commandes [BACKUP CFSTRUCT](#) et [RECOVER CFSTRUCT](#).

### Concepts associés

«Définition des options de déchargement pour les messages partagés», à la page 182

Vous pouvez choisir l'emplacement de stockage des données d'un message de file d'attente partagée : dans une table Db2 ou dans un fichier de messages partagé. Vous pouvez également sélectionner les messages qui sont déchargés en fonction de leur taille et de l'utilisation actuelle de la structure d'unité de couplage.

«Gestion de l'environnement de fichiers de messages partagés», à la page 185

Si vous sélectionnez des fichiers de messages partagés pour décharger les messages de grande taille, vous devez également connaître les informations qu'IBM MQ utilise pour gérer ces fichiers et les commandes permettant d'utiliser ces informations. Consultez cette rubrique pour comprendre comment gérer les fichiers de messages partagés.

### Définition des options de déchargement pour les messages partagés

Vous pouvez choisir l'emplacement de stockage des données d'un message de file d'attente partagée : dans une table Db2 ou dans un fichier de messages partagé. Vous pouvez également sélectionner les messages qui sont déchargés en fonction de leur taille et de l'utilisation actuelle de la structure d'unité de couplage.

Vous pouvez décharger les données de message des files d'attente partagées à partir de l'unité de couplage et les stocker dans une table Db2 ou dans un fichier géré IBM MQ appelé *fichier de messages partagé* (SMDS).

Pour les messages plus volumineux que la taille des entrées de l'unité de couplage (63 ko), le déchargement des données de message dans un fichier de messages partagés plutôt que dans Db2 peut permettre d'améliorer nettement les performances.

Chaque message de la file d'attente partagée est géré à l'aide d'une entrée de liste de la structure de l'unité de couplage, mais lorsque les données de message sont déchargées dans le fichier de messages partagé, l'entrée de l'unité de couplage contient uniquement les informations de contrôle et une liste des références aux blocs disque dans lesquels le message est stocké. Lorsque vous utilisez ce mécanisme, le volume de stockage de l'élément de l'unité de couplage requis pour chaque message ne représente qu'une fraction de sa taille réelle.

## Sélection de l'emplacement de stockage des messages de la file d'attente

La sélection du stockage des messages partagé SMDS ou Db2 est contrôlée avec le paramètre **OFFLOAD(SMDS|DB2)** de la définition **CFSTRUCT**. **OFFLOAD(SMDS)** est la valeur par défaut.

Ce paramètre requiert également que **CFSTRUCT** utilise **CFLEVEL(5)** ou une version ultérieure. Seuls les gestionnaires de files d'attente de niveau IBM WebSphere MQ 7.1 ou d'un niveau supérieur peuvent se connecter à une structure de l'unité de couplage à ce niveau.

Il n'est possible de modifier une structure jusqu'à **CFLEVEL(5)** que si tous les gestionnaires de files d'attente du groupe de partage de files d'attente se trouvent à la version IBM WebSphere MQ 7.1 ou ultérieure.

Le paramètre **OFFLOAD** est valide uniquement depuis **CFLEVEL(5)**. Pour plus d'informations, voir DEFINE CFSTRUCT.

**OFFLOAD(DB2)** est pris en charge principalement à des fins de migration.

## Sélection des messages de la file d'attente partagée à décharger

Les données de message sont déchargées vers le fichier de messages partagés ou Db2 en fonction de leur taille et de l'utilisation actuelle de la structure d'unité de couplage. Il existe trois règles. Chacune d'entre elles définit une paire de correspondance de paramètres. Ces paramètres représentent un pourcentage de seuil d'utilisation de la structure d'unité de couplage correspondant (**OFFLDnTH**) et une limite de taille de message (**OFFLDnSZ**).

La mise en oeuvre actuelle de ces trois règles est définie par les paires de mots clés suivantes :

- OFFLD1TH et OFFLD1SZ
- OFFLD2TH et OFFLD2SZ
- OFFLD3TH et OFFLD3SZ

Paire de règles	Valeur par défaut	Description
Paire de règles 1	OFFLD1TH(70) et OFFLD1SZ(32K)	Si la structure d'unité de couplage est saturée à plus de 70%, décharge les données des messages dont la taille dépasse 32 ko
Paire de règles 2	OFFLD2TH(80) et OFFLD2SZ(4K)	Si la structure d'unité de couplage est saturée à plus de 80%, décharge les données des messages dont la taille dépasse 4 ko
Paire de règles 3	OFFLD3TH(90) et OFFLD3SZ(0K)	Si la structure d'unité de couplage est saturée à plus de 90%, décharge les données des messages dont la taille dépasse 0 ko (tous les messages)

Si une règle de déchargement a la valeur OFFLD x SZ de 64K, cela indique que la règle n'est pas en vigueur. Dans ce cas, les messages ne seront déchargés que si une autre règle de déchargement est en vigueur, ou si le message est supérieur à 63,75 Ko et donc trop grand pour être stocké dans la structure.

Chaque message déchargé requiert toujours 0,75 ko de stockage dans l'unité de couplage.

Les trois règles de déchargement qui peuvent être définies pour chaque structure doivent être utilisées comme suit.

- Performance

- Lorsque l'espace disponible est largement suffisant dans la structure applicative, les données de message sont déchargées uniquement si elles sont trop volumineuses pour y être stockées ou si leur taille dépasse le seuil inférieur de taille des messages en-deçà duquel la valeur liée au stockage des données dans la structure ne justifie pas l'espace nécessaire à ce stockage.
- Si un seuil spécifique de taille des messages est nécessaire, il est, par convention, défini à l'aide de la première règle de déchargement.
- Capacité
  - Lorsque l'espace disponible dans la structure applicative est peu important, la plus grande quantité possible de données de message doit être déchargée afin d'optimiser l'utilisation de l'espace restant.
  - Par convention, la troisième règle de déchargement permet d'indiquer que lorsque la structure est pratiquement saturée, la plupart des messages doivent être déchargés. Les entrées de la structure applicative auront donc, en règle générale, la taille minimale (environ 0,75 ko).
  - Le paramètre du seuil d'utilisation doit être choisi en fonction de la taille de la structure applicative et de la file de messages en attente maximale anticipée. Par exemple, si cette file contient un million de messages, l'espace de structure requis pour ce nombre de messages est d'environ 0,75 Go. Cela signifie que si la taille de la structure est d'environ à 10 Go, le seuil d'utilisation de déchargement de tous les messages doit être égal à 92 % au plus.
  - L'espace de la structure est divisé en éléments et en entrées. Même si l'espace total est suffisant, l'une de ces deux parties peut manquer d'espace avant l'autre. Le système propose les fonctions AUTOALTER, qui permettent d'ajuster ce rapport lorsque c'est nécessaire. Cette fonction n'est toutefois pas très sensible et l'espace réellement disponible peut être parfois inférieur. Il est donc recommandé de ne pas utiliser plus de 90 % de l'espace maximal de la structure. Dans l'exemple précédent, le seuil d'utilisation de déchargement de tous les messages serait ainsi fixé à environ 80 %.
- Transition avec marge de sécurité
  - Lorsque l'espace disponible dans la structure d'unité de couplage diminue, un soudain changement des caractéristiques des performances n'est pas souhaitable. Un changement subit du rapport habituel entre les entrées et les éléments n'est également pas recommandé pour la gestion de l'unité de couplage.
  - Par convention, la deuxième règle de déchargement offre une marge de sécurité entre les règles de déchargement pondérées de capacité et de performances. Elle permet d'augmenter de manière importante l'activité de déchargement lorsque l'espace utilisé dans la structure d'unité de couplage dépasse un seuil intermédiaire. Cela signifie que l'espace restant est utilisé plus lentement et que le processus de modification automatique de l'unité de couplage bénéficie de plus de temps pour s'adapter à des niveaux d'utilisation supérieurs.

S'il est impossible d'étendre la structure de l'unité de couplage et si un certain nombre au moins de messages prédéterminés doivent être stockés, la troisième règle peut être si nécessaire modifiée de sorte que le déchargement des données de tous les messages démarre à partir d'un seuil approprié et que l'espace nécessaire soit réservé à ces messages.

Si, par exemple, la structure de l'unité de couplage est de 4 Go et que le nombre prédéterminé de messages est égal à 1 million,  $1\ 000\ 000 \times 0,75\ \text{Ko}$  sont nécessaires, c'est-à-dire 768 Mo ou 18,75 % de 4 Go. Dans ce cas, le seuil de déchargement de tous les messages doit être défini autour de 80 % plutôt que 90 %. Les paramètres obtenus sont OFFLD3TH(80) et OFFLD3SZ(0K). Les autres paramètres de déchargement doivent alors être ajustés.

Si le déchargement des messages de très petite taille a un impact important sur les performances, mais que l'impact relatif du déchargement des messages de plus grande taille soit inférieur, vous pouvez réduire les seuils d'utilisation des autres règles pour décharger en priorité les messages de grande taille. L'espace laissé aux messages de petite taille avant leur déchargement sera ainsi plus important.

Par exemple, si des messages dont la taille dépasse 32 ko sont souvent émis, mais que le temps écoulé pour leur déchargement (déterminé par les statistiques RMF ou les performances de

l'application) est pratiquement identique au temps nécessaire à leur conservation dans l'unité de couplage, le seuil de la première règle peut être défini à 0 % afin de décharger tous les messages de ce type. Les paramètres obtenus sont alors OFFLD1TH(0) et OFFLD1SZ(32K). Dans ce cas également, les autres paramètres de déchargement doivent alors être ajustés.

Si les messages de taille intermédiaire (par exemple 6 ou 16 ko) sont nombreux, il peut être utile de modifier l'option de taille des messages pour la deuxième règle, de sorte que les messages les plus volumineux soient déchargés à un seuil d'utilisation relativement bas. Une importante quantité d'espace pourrait ainsi être gagnée et les messages de plus petite taille seraient stockés uniquement dans l'unité de couplage.

**z/OS**

## **Gestion de l'environnement de fichiers de messages partagés**

Si vous sélectionnez des fichiers de messages partagés pour décharger les messages de grande taille, vous devez également connaître les informations qu'IBM MQ utilise pour gérer ces fichiers et les commandes permettant d'utiliser ces informations. Consultez cette rubrique pour comprendre comment gérer les fichiers de messages partagés.

### **Objets SMDS**

Les propriétés et le statut de chaque fichier de messages partagé sont suivis dans un objet SMDS partagé pouvant être mis à jour via n'importe quel gestionnaire de files d'attente du groupe de partage.

Un fichier de messages partagé par gestionnaire de files d'attente peut accéder à chaque structure d'unité de couplage. Le fichier de messages partagé est identifié par le nom du gestionnaire de files d'attente propriétaire, défini à l'aide du mot clé SMDS, et par le nom de la structure, défini à l'aide du mot clé CFSTRUCT.

**Remarque :** lorsque vous définissez des fichiers de messages partagés (SMDS) pour une structure, vous devez en définir un pour chaque gestionnaire de files d'attente.

L'objet SMDS est stocké dans un tableau (avec une entrée par gestionnaire de files d'attente du groupe) qui forme une extension de l'objet CFSTRUCT correspondant stocké dans Db2.

Aucune commande ne permet de définir ou supprimer l'objet SMDS car il est créé ou supprimé dans le cadre de l'objet CFSTRUCT, mais une commande permet de le modifier et de modifier les paramètres d'un gestionnaire de files d'attente propriétaire donné.

Pour plus d'informations sur les commandes associées aux fichiers de messages partagés, voir [«Commandes associées aux fichiers de messages partagés», à la page 197](#)

### **Informations SMDSCONN**

Il peut arriver qu'un fichier de messages partagé soit dans un état normal, mais qu'un ou plusieurs gestionnaires de files d'attente ne parviennent pas à s'y connecter, par exemple en raison d'un problème lié à une définition de sécurité ou à la connectivité d'un disque à accès direct. Chaque gestionnaire de files d'attente doit donc assurer un suivi du statut de la connexion et des informations de disponibilité concernant chaque fichier de messages partagé. Vous saurez ainsi si la connexion est possible, et, dans le cas contraire, pourquoi.

Les informations SMDSCONN représentent une connexion entre un gestionnaire de files d'attente et un fichier de messages partagé. Ce fichier est identifié par une combinaison du gestionnaire de files d'attente qui en est le propriétaire (comme défini par le mot clé SMDS de l'objet partagé) et le nom CFSTRUCT.

Aucun paramètre ne permet d'identifier le gestionnaire de files d'attente qui se connecte car les commandes adressées à un gestionnaire donné peuvent uniquement faire référence aux informations SMDSCONN relatives à ce gestionnaire.

Les entrées d'informations SMDSCONN sont gérées dans la mémoire principale du gestionnaire de files d'attente propriétaire et sont recrées lors de son redémarrage. Toutefois, si une connexion établie à partir d'un gestionnaire de files d'attente donné a été interrompue de manière explicite, ces informations sont également stockées sous la forme d'un indicateur dans un tableau de connexions de l'objet

CFSTRUCT ou SMDS correspondant. Elles sont ainsi conservées en cas de redémarrage du gestionnaire de files d'attente.

## Informations de statut et de disponibilité

Les informations de statut indiquent le statut d'une ressource ou d'une connexion (par exemple si elle est actuellement inutilisée, si elle est normalement utilisée ou si elle doit faire l'objet d'une récupération). Ces informations sont généralement décrites à l'aide du mot clé STATUS. Les valeurs possibles dépendent du type d'objet.

En temps normal, les informations de statut sont mises à jour automatiquement, par exemple lorsqu'une erreur est détectée lors de l'utilisation d'une ressource ou d'une connexion. Dans certains cas, il est cependant possible d'utiliser une commande pour mettre à jour le statut, par exemple lorsqu'un gestionnaire de files d'attente ne parvient pas à l'identifier automatiquement.

Les informations de disponibilité indiquent s'il est possible d'utiliser la ressource ou la connexion. Elles sont principalement déterminées par les informations de statut. Il existe trois niveaux de disponibilité pour les types de ressource ou de connexion utilisés dans le cadre de la prise en charge des fichiers de messages partagés :

### Disponible

Signifie que la ressource est disponible en vue d'une utilisation normale. Cela n'implique pas que cette ressource soit actuellement en cours d'utilisation (l'utilisation ou non de la ressource est déterminée par la valeur STATUS). Dans le cas d'un fichier pour lequel une opération de redémarrage est nécessaire, le gestionnaire de files d'attente propriétaire peut ouvrir le fichier, mais les autres gestionnaires de files d'attente doivent attendre jusqu'à la restauration de l'état ACTIVE du fichier.

### Indisponible en raison d'une erreur

Signifie que la ressource est devenue automatiquement non disponible en raison d'une erreur et qu'elle ne sera plus disponible tant qu'une opération de réparation ou de récupération n'aura pas été effectuée. Toutefois, les opérations visant à rétablir sa disponibilité sont autorisées sans intervention de l'opérateur. Elles peuvent être déclenchées par une commande visant à marquer la ressource comme activée ou par une commande permettant de modifier son statut et d'indiquer que la récupération a été exécutée.

La raison de la non disponibilité de la ressource est généralement indiquée de manière évidente dans la valeur STATUS associée. Dans certains cas, cependant, d'autres raisons peuvent être à l'origine de cet état. Une valeur REASON est alors fournie pour indiquer ces raisons.

### Indisponible en raison d'une commande de l'opérateur

Signifie que l'accès à la ressource a été désactivé de manière explicite par une commande. Seule une commande permettant de réactiver la disponibilité peut rétablir celle-ci.

### Disponibilité SMDS

La disponibilité de l'objet SMDS partagé est décrite par le mot clé ACCESS et par les valeurs qui peuvent lui être associées : ENABLED, SUSPENDED et DISABLED.

Pour mettre à jour la disponibilité de l'objet concerné, utilisez la commande **RESET SMDS** à partir de n'importe quel gestionnaire de files d'attente du groupe de partage pour définir ACCESS(ENABLED) ou ACCESS(DISABLED).

Si le statut de la disponibilité était précédemment ACCESS(SUSPENDED), remplacez-le par ACCESS(ENABLED) pour déclencher une nouvelle tentative d'utilisation du fichier de messages partagé. Si l'erreur précédemment observée est toujours présente, le statut ACCESS(SUSPENDED) est restauré.

### Disponibilité SMDSCONN

La disponibilité des entrées d'informations SMDSCONN locales est décrite par le mot clé AVAIL et par les valeurs qui peuvent lui être associées : NORMAL, ERROR ou STOPPED. La disponibilité peut être mise à jour à l'aide d'une commande **START SMDSCONN** ou **STOP SMDSCONN** adressée à un gestionnaire de files d'attente spécifique pour activer ou désactiver sa connexion.

Si le statut de la disponibilité était précédemment AVAIL(ERROR), remplacez-le par AVAIL(NORMAL) pour déclencher une nouvelle tentative d'utilisation du fichier de messages partagé. Si l'erreur précédemment observée est toujours présente, le statut AVAIL(ERROR) est restauré.

## Statut et disponibilité partagés du fichier de messages partagé

La disponibilité de chaque fichier de messages partagé est gérée au sein du groupe à l'aide des informations de statut partagé, qui peuvent être affichées à l'aide de la commande **DISPLAY CFSTATUS** avec TYPE (SMDS). Cette commande permet d'afficher des informations de statut pour chaque gestionnaire de files d'attente ayant activé un fichier pour chaque structure. Chaque fichier peut prendre l'un des états suivants :

### NOTFOUND

Le fichier correspondant n'a pas encore été activé. Ce statut apparaît uniquement lorsqu'un gestionnaire de files d'attente est spécifié, car les fichiers non activés sont ignorés lorsque tous les gestionnaires de files d'attente sont sélectionnés.

### NEW

Le fichier est ouvert et initialisé pour la première fois, prêt à devenir actif.

### ACTIF

Le fichier est complètement disponible. Il doit être alloué et ouvert par tous les gestionnaires de files d'attente de la structure.

### EN ECHEC

Le fichier n'est pas du tout disponible (sauf pour une opération de récupération). Il doit être fermé et libéré par tous les gestionnaires de files d'attente.

### INRECOVER

La récupération des supports (à l'aide de RECOVER CFSTRUCT) est en cours pour ce fichier.

### RECOVERED

Indique qu'une commande a été émise pour basculer un fichier ayant échoué à l'état actif, mais l'opération de redémarrage nécessaire n'est pas encore terminée et le fichier peut uniquement être ouvert par le gestionnaire de files d'attente propriétaire.

### EMPTY

Le fichier ne contient aucun message. Cet état est attribué au fichier si le gestionnaire de files d'attente qui en est le propriétaire le ferme normalement à un moment où il ne contient aucun message. Le fichier peut également être mis à l'état EMPTY lorsque son contenu précédent doit être détruit car la structure a été vidée (à l'aide de la commande **RECOVER CFSTRUCT** avec TYPE PURGE ou, dans le cas d'une structure non récupérable, en supprimant l'instance précédente de celle-ci). Lors de la prochaine ouverture du fichier par le gestionnaire de files d'attente qui en est propriétaire, la mappe d'espace redevient vide et son statut devient ACTIVE. Son contenu précédent n'étant plus nécessaire, le fichier se trouvant dans cet état peut être remplacé par un fichier récemment alloué, par exemple pour modifier l'allocation d'espace ou pour le déplacer vers un autre volume.

Le résultat de la commande contient la date et l'heure à laquelle la consignation de la récupération a été activée et, le cas échéant, la date et l'heure de la défaillance du fichier, s'il n'est plus actif.

Un fichier de messages partagé peut être mis à l'état FAILED à l'aide d'une commande **RESET SMDS** ou automatiquement lorsque l'un des types d'erreur suivants est détecté:

- Le gestionnaire de files d'attente qui en est propriétaire ne peut ni allouer ni ouvrir le fichier.
- Le fichier a été ouvert par l'un des gestionnaires de files d'attente mais la validation de son en-tête échoue.
- Une erreur d'E-S permanente se produit lorsque le gestionnaire de files d'attente propriétaire lit ou écrit les données.
- Une erreur d'E-S permanente se produit lorsqu'un autre gestionnaire de files d'attente lit les données à partir d'un fichier qui a exécuté les tâches de traitement ouvert et de validation.

Lorsque l'état d'un fichier est FAILED ou INRECOVER, il n'est pas disponible en vue d'une utilisation normale. Si son état de disponibilité est ACCESS(ENABLED), cet état passe alors à ACCESS(SUSPENDED).

Si un fichier a été mis à l'état FAILED mais qu'aucune récupération de support ne soit nécessaire, par exemple parce que les données étaient toujours valides mais que l'unité de stockage était temporairement hors ligne, la commande **RESET SMDS** permet de faire passer le statut directement à l'état RECOVERED.

Lorsque le fichier passe à l'état RECOVERY, soit à la fin du processus de récupération, soit à la suite de la commande **RESET SMDS**, il est prêt à être utilisé à nouveau une fois le processus de redémarrage terminé. Si l'état du fichier était ACCESS(SUSPENDED), l'état ACCESS(ENABLED) est automatiquement restauré, ce qui permet au gestionnaire de files d'attente propriétaire de procéder au traitement du redémarrage. Une fois cette opération terminée, l'état passe à ACTIVE et tous les autres gestionnaires de files d'attente peuvent à nouveau se connecter au fichier.

## Statut et disponibilité de la connexion des fichiers de messages partagés

Chaque gestionnaire de files d'attente gère des informations sur le statut et la disponibilité locaux de ses connexions à chaque fichier de messages partagé dont il est le propriétaire où dont d'autres gestionnaires sont les propriétaires. Vous pouvez afficher ces informations via la commande **DISPLAY SMDSCONN**.

S'il ne parvient pas à accéder à un fichier de messages partagé à l'état ACTIVE appartenant à un autre gestionnaire de files d'attente, il marque la connexion comme étant indisponible de son point de vue.

Si l'erreur indique clairement un problème lié au fichier lui-même, le gestionnaire de files d'attente modifie automatiquement le statut partagé. Le fichier apparaît alors comme étant à l'état FAILED. Toutefois, si l'erreur est due à un problème d'environnement (par exemple s'il n'est pas autorisé à ouvrir le fichier), le gestionnaire de files d'attente émet des messages d'erreur et considère le fichier comme étant non disponible, mais il ne modifie pas son statut. Si le problème d'environnement est en réalité un problème lié au fichier, (par exemple s'il a été alloué sur une unité à laquelle certains des gestionnaires de files d'attente n'ont pas accès), un opérateur peut utiliser la commande **RESET SMDS** en indiquant **STATUS(FAILED)** pour permettre la récupération ou la réparation du fichier, si nécessaire.

Si la connexion au fichier de messages partagé n'a pas pu être établie, mais que ce fichier semble valide, une nouvelle tentative peut être déclenchée par la commande **START SMDSCONN** pour le gestionnaire de files d'attente propriétaire.

En cas de nécessité fonctionnelle de mettre temporairement fin à la connexion entre un gestionnaire de files d'attente donné et un fichier alors que celui-ci n'est pas endommagé, il est possible de fermer le fichier et de le libérer à l'aide de la commande **STOP SMDSCONN**. Si le fichier est en cours d'utilisation, le gestionnaire de files d'attente le ferme normalement (mais toutes les demandes concernant des données contenues dans ce fichier seront rejetées avec un code retour). S'il s'agit du fichier appartenant au gestionnaire de files d'attente, celui-ci sauvegarde la mappe d'espace au cours du traitement de la commande **CLOSE**. Un redémarrage est donc inutile.

Si un fichier doit temporairement être mis hors service par tous les gestionnaires de files d'attente (par exemple pour être déplacé) alors qu'il n'est pas endommagé, il est préférable de commencer par utiliser la commande **STOP SMDSCONN** avec l'option **CMDScope(\*)** pour arrêter les gestionnaires de files d'attente qui l'utilisent. Il sera alors inutile de procéder à une opération de redémarrage lorsque le fichier sera remis en service. En revanche, le statut FAILED du fichier indique aux gestionnaires de files d'attente qu'ils doivent immédiatement arrêter de l'utiliser. La mappe d'espace ne sera pas sauvegardée et elle devra être régénérée par une opération de redémarrage.

L'accès à un fichier de messages partagé précédemment à l'état ACCESS(SUSPENDED) fera l'objet d'une nouvelle tentative si le gestionnaire de files d'attente est redémarré.

## Journalisation de la récupération des fichiers de messages partagés

Les messages partagés persistants sont consignés à des fins de récupération de support. Cela signifie qu'il est possible de récupérer ces messages après la défaillance d'une structure d'unité de couplage ou d'un fichier de messages partagé, à condition que les journaux de reprise soient toujours intacts. Les messages persistants peuvent également être recréés à partir des journaux de reprise d'un autre site dans le cadre d'une reprise après incident.

Lorsque les données de message sont enregistrées dans un fichier de messages partagé, chaque bloc enregistré est consigné séparément et suivi de l'entrée de message (ainsi que de la mappe de données) enregistrée dans l'unité de couplage. La procédure de récupération récupère toujours la structure de l'unité de couplage, mais elle n'a pas besoin de récupérer chaque fichier de messages partagés sauf lorsqu'un tel fichier est à l'état FAILED ou qu'il est à l'état ACTIVE, mais que l'enregistrement de l'en-tête de fichier n'est plus valide car le fichier a été recréé. Un fichier n'est pas sélectionné en vue de sa récupération si son statut est ACTIVE et que son en-tête est toujours valide, ni si son statut est EMPTY, ce qui signifie qu'aucun message n'y était stocké lors de l'incident.

## Sauvegardes des fichiers de messages partagés

Lorsque la commande **BACKUP CFSTRUCT** est utilisée pour effectuer une sauvegarde des messages partagés d'une structure applicative, les données relatives aux messages persistants figurant dans les fichiers de messages partagés sont sauvegardées en même temps, de même que les données relatives aux messages partagés persistants précédemment stockés dans la base de données.

## Récupération des fichiers de messages partagés

Si un fichier de messages partagé est endommagé ou perdu, il doit être mis à l'état FAILED. Les gestionnaires de files d'attente ne l'utiliseront alors plus jusqu'à ce qu'il soit réparé. Cette procédure se fait normalement de manière automatique, mais elle peut également être exécutée à l'aide de la commande **RESET SMDS** en indiquant STATUS(FAILED).

Si le fichier de messages partagé contient des messages persistants, il est possible de les récupérer à l'aide de la commande **RECOVER CFSTRUCT**. Cette commande commence par restaurer les données de message persistant de ce fichier de messages partagé à partir de la commande **BACKUP CFSTRUCT** la plus récente, puis applique les modifications consignées depuis l'exécution de cette commande. Si aucune commande **BACKUP CFSTRUCT** n'a été exécutée depuis la première activation du fichier, celui-ci est réinitialisé à l'état vide et toutes les modifications apportées depuis l'activation sont vidées.

Si le contenu **CFSTRUCT** et tous les fichiers de messages partagés ne sont pas disponibles, par exemple en cas de reprise après incident, ils peuvent tous être récupérés à l'aide d'une seule commande **RECOVER CFSTRUCT**.

Si un fichier de messages partagé est endommagé alors que la récupération n'était pas active pour **CFSTRUCT** ou que le journal contenant le dernier **BACKUP CFSTRUCT** est indisponible ou inutilisable, il est impossible de récupérer les messages déchargés dans ce fichier. Dans ce cas, vous pouvez utiliser la commande **RECOVER CFSTRUCT** avec le paramètre **TYPE(PURGE)** pour marquer le fichier de messages partagé comme vide et supprimer les messages de la structure dont certaines données étaient stockées dans ce fichier.

Lorsque la commande **RECOVER CFSTRUCT** est émise, le statut du fichier de messages partagés passe de FAILED à **INRECOVER**. Si la récupération aboutit, ce statut passe automatiquement à **RECOVERED**. Dans le cas contraire, il repasse à FAILED.

La restauration de l'état **RECOVERED** du fichier indique au gestionnaire de files d'attente qu'il peut désormais tenter d'ouvrir le fichier et d'exécuter l'opération de redémarrage.

## Récupération et points de synchronisation des fichiers de messages partagés

Le processus de récupération des fichiers de messages partagés réapplique les modifications pour tous les enregistrements de journal complets jusqu'à la fin du journal, indépendamment des points de synchronisation.

Si des modifications ont été apportées dans les points de synchronisation, l'opération de redémarrage ou de récupération pour **CFSTRUCT** risque d'entraîner l'annulation des demandes non validées. Il se peut donc que certaines modifications récupérées ne soient pas utilisées, mais leur récupération ne présente aucun risque.

Il est également possible qu'un message **MQPUT** non validé ait été consigné dans la structure mais que les données correspondantes n'aient pas été consignées dans le fichier ou dans le journal (l'exécution des

E-S n'est en effet forcée qu'au début du traitement des points de synchronisation). Cette procédure ne présente aucun risque car l'opération de redémarrage va annuler l'entrée du message dans la structure. Peu importe, donc, qu'il s'agisse de données non récupérées.

## Opération de redémarrage des fichiers de messages partagés

Si la connexion d'un gestionnaire de files d'attente à un CFSTRUCT s'interrompt normalement, le gestionnaire de files d'attente consigne la mappe d'espace du bloc libre pour chaque fichier de messages partagé dans une zone de point de contrôle du fichier juste avant la fermeture de celui-ci. Il est alors possible de relire cette mappe lors de la restauration de la connexion, à condition qu'aucune opération de reprise ne soit nécessaire pour CFSTRUCT ou le fichier de messages partagé avant le redémarrage suivant.

Toutefois, si un gestionnaire de files d'attente s'arrête de façon anormale ou si la structure ou le fichier requiert une opération de reprise, d'autres opérations sont nécessaires pour régénérer dynamiquement la mappe d'espace lors du rétablissement de la connexion avec la structure.

La récupération du fichier n'étant pas nécessaire, le redémarrage du gestionnaire de files d'attente analyse simplement le contenu actuel de la structure pour localiser les références aux données de message appartenant au gestionnaire de files d'attente actuel et marque les blocs de données concernés comme appartenant à ce gestionnaire dans la mappe d'espace. Les autres gestionnaires de files d'attente continuent à utiliser la structure et lisent les données appartenant au gestionnaire de files d'attente en cours de redémarrage alors que l'espace de mappe est en cours de régénération.

## Redémarrage des fichiers de messages partagés après la récupération

Si un fichier de messages partagé a été récupéré d'une sauvegarde, tous les messages non persistants sont perdus et si le fichier a été récupéré à l'aide de la commande TYPE(PURGE), tous les messages stockés dans le fichier sont perdus. Tant que la récupération n'est pas terminée, le fichier est marqué comme FAILED ou INRECOVER et toute tentative de lire l'un des messages concernés à partir d'un gestionnaire de files d'attente renvoie un code d'erreur indiquant que le fichier est temporairement indisponible.

Une fois le fichier récupéré, son statut passe à RECOVERED. Le gestionnaire de files d'attente qui en est propriétaire peut alors l'ouvrir en vue de son redémarrage, mais le fichier reste indisponible pour les autres gestionnaires de files d'attente. Le redémarrage du gestionnaire de files d'attente analyse la structure afin de régénérer la mappe d'espace pour les messages restants. Les messages ayant perdu des données sont également recherchés, puis supprimés de la structure (ou marqués comme perdus afin d'être supprimés ultérieurement).

Le statut du fichier passe automatiquement de RECOVERED à ACTIVE lorsque l'analyse est terminée. Les autres gestionnaires de files d'attente peuvent alors recommencer à l'utiliser.

## Informations sur l'utilisation des fichiers de messages partagés

La commande DISPLAY USAGE affiche désormais des informations sur l'utilisation de l'espace et du pool de mémoire tampon de tous les fichiers de messages partagés actuellement ouverts. Ces informations s'affichent si la nouvelle option TYPE(SMDS) ou l'option existante TYPE(ALL) est spécifiée.

## Considérations liées aux performances et à la capacité des données de messages partagés

### Surveillance de l'utilisation des fichiers

Pour afficher le pourcentage complet de chaque fichier de messages partagés détenu par le gestionnaire de files d'attente, exécutez la commande **DISPLAY USAGE** avec l'option **TYPE(SMDS)**.

Normalement, le gestionnaire de files d'attente étend automatiquement un fichier de messages partagé lorsqu'il est plein à 90 %, à condition que l'option **DSEXPAND(YES)** soit appliquée pour la définition SMDS. Tel est le cas lorsque l'option SMDS a pour valeur **DSEXPAND(YES)** ou **DSEXPAND(DEFAULT)** et que l'option par défaut CFSTRUCT a pour valeur **DSEXPAND(YES)**.

Si la tentative d'extension échoue car aucune taille d'allocation secondaire n'a été indiquée lors de la création du fichier (le message IEC070I et le code anomalie 203 sont alors émis), le gestionnaire de files d'attente renouvelle la demande d'extension en mentionnant une allocation secondaire de remplacement d'environ 20% de la taille actuelle.

Lors de l'extension d'un fichier, les nouvelles extensions sont formatées au cours de cette opération, ce qui peut prendre plusieurs dizaines de secondes, voire plusieurs minutes pour des extensions très importantes. Une fois le formatage terminé, le nouvel espace devient disponible et le catalogue est mis à jour avec la nouvelle valeur utilisée pour l'intervalle de contrôle.

Si de nouveaux messages sont créés rapidement, le fichier de données risque d'être saturé avant la fin du processus d'extension. Dans ce cas, toute demande d'allocation d'espace ayant échoué est temporairement suspendue jusqu'à ce que la tentative d'extension aboutisse et que le nouvel espace soit créé. Si l'extension aboutit, la tentative est automatiquement relancée.

Si une tentative d'extension échoue car l'espace disponible est insuffisant ou car le nombre maximal d'extensions a déjà été atteint, un message indiquant le motif de l'échec est émis, puis la valeur de l'option de substitution pour le service SMDS concerné devient automatiquement **DSEXPAND(NO)** pour empêcher toute autre tentative d'extension. Dans ce cas, le fichier risque d'être saturé. D'autres opérations sont alors nécessaires, comme décrit à la section [Le fichier est saturé](#).

### Surveillance de l'utilisation de la structure applicative

Le niveau d'utilisation d'une structure d'application peut être affiché à l'aide de la commande MVS **DISPLAY XCF, STRUCTURE** en spécifiant le nom complet de la structure d'application (y compris le préfixe de groupe de partage de files d'attente). Le message de réponse IXC360I présente l'utilisation actuelle des éléments et des entrées.

Lorsque l'utilisation de la structure dépasse la valeur **FULLTHRESHOLD** définie dans les règles CFRM, le système émet le message IXC585E et il peut exécuter des actions **ALTER** automatique si cela a été indiqué, ce qui risque de modifier le rapport élément-entrée ou d'augmenter la taille de la structure.

### Optimisation des tailles de mémoire tampon

Dans un pool de mémoire tampon partagé, chaque mémoire tampon permet de lire ou d'écrire une plage contiguë de pages pour un même message pouvant atteindre la taille du bloc logique. Si le message déborde sur d'autres blocs, les plages de pages de ces blocs requièrent une mémoire tampon distincte.

Les mémoires tampons contenant les données de message consécutives à une opération d'écriture ou de lecture sont conservées en mémoire et réutilisées à l'aide du schéma de la mémoire cache la moins récemment utilisé. Au cas où une demande de lecture des mêmes données serait émise peu de temps après, aucun accès au disque ne sera nécessaire. Cette fonction permet donc d'optimiser considérablement les opérations d'écriture, puis de lecture des messages partagés par des applications s'exécutant sur le même système. Lorsque des messages appartenant à un autre gestionnaire de files d'attente sont recherchés en vue de leur sélection, puis récupérés, elle permet également d'éviter de devoir relire le message à partir du disque.

Cela signifie que le nombre de mémoires tampons requises pour chaque structure applicative est égal à un pour chaque demande d'API simultanée lisant ou écrivant des messages volumineux pour cette structure, plus un certain nombre de mémoires tampons supplémentaires qui seront utilisées pour sauvegarder les données ayant fait l'objet d'un accès récent afin d'optimiser les accès en lecture ultérieurs.

Pour les pools de mémoire tampon partagés, si le nombre de mémoires tampons est insuffisant, les demandes d'API se contentent d'attendre si aucune mémoire tampon n'est disponible immédiatement. Toutefois, ce cas de figure doit être évité, car il peut être à l'origine d'une dégradation importante des performances.

Les statistiques de la commande **DISPLAY USAGE** pour les pools de mémoire tampon partagés indiquent s'il y a eu des attentes de mémoire tampon dans l'intervalle de statistiques en cours, et affichent également le nombre le plus faible de mémoires tampon libres (ou une valeur négative indiquant le nombre maximal d'unités d'exécution qui ont attendu une mémoire tampon à tout moment), le nombre de mémoires tampons qui ont enregistré des données et le pourcentage de

fois où une demande de mémoire tampon a trouvé des données sauvegardées sur la chaîne LRU ("LRU hits") au lieu de devoir les lire ("LRU misses")<sup>1</sup>.

- S'il y a eu des attentes, le nombre de mémoires tampons doit être augmenté.
- Si plusieurs mémoires tampons sont inutilisées, il est possible d'en réduire le nombre. La mémoire disponible pour d'autres opérations dans la région sera ainsi supérieure.
- Si plusieurs mémoires tampons contiennent des données sauvegardées mais que la proportion d'opérations de lecture de ces données ayant abouti est infime, le nombre de mémoires tampons peut être réduit, ces mémoires pouvant être mieux utilisées à d'autres fins. Le nombre de mémoires tampons ne doit toutefois pas descendre en-deçà du nombre minimal de mémoires tampons disponibles, sans quoi des attentes pourraient être déclenchées, et il doit de préférence rester suffisamment élevé pour que le nombre minimal de mémoires tampons disponibles reste bien supérieur à zéro.

## Suppression des fichiers de messages partagés

La commande DELETE CFSTRUCT (qui n'est autorisée que lorsque toutes les files d'attente partagées de la structure sont vides et fermées) ne supprime pas les fichiers de messages partagés eux-mêmes, mais peut être supprimée de la manière habituelle après l'exécution de cette commande. Si le même fichier doit être réutilisé en tant que fichier de messages partagé, il doit d'abord être reformaté, puis ramené à l'état vide.

## Situations d'exception pour les fichiers de messages partagés

Un certain nombre de situations d'exception peuvent se produire dans des circonstances normales, même si aucune erreur logicielle ou matérielle ne s'est produite.

### Le fichier est saturé

Si un fichier est saturé, mais qu'il est impossible de l'étendre ou que la tentative d'extension échoue, les applications utilisant le gestionnaire de files d'attente correspondant pour écrire des messages de grande taille dans la structure applicative correspondante reçoivent l'erreur 2192, MQRD\_STORAGE\_MEDIUM\_FULL (également exprimée sous la forme MQRD\_PAGESET\_FULL).

Un fichier peut être saturé en cas de défaillance de l'application censée traiter les données, ce qui entraîne l'accumulation de nombreux messages en attente. Dans un tel cas de figure, l'extension du fichier ne constitue qu'une solution temporaire et il est important de rétablir dès que possible le bon fonctionnement de l'application.

Si l'espace nécessaire a pu être libéré, vous pouvez utiliser la commande **ALTER SMDS DSEXPAND(YES)** ou **DSEXPAND(DEFAULT)** (la valeur YES est alors définie ou considérée comme la valeur associée par défaut à la commande **DSEXPAND** pour la définition CFSTRUCT) afin de déclencher une nouvelle tentative. Cependant, si l'échec est dû au fait que le nombre maximal d'extensions a été atteint, la nouvelle tentative d'extension sera rejetée avec un message et **DSEXPAND(NO)** sera à nouveau défini. Dans ce cas, la seule manière d'étendre le fichier consiste à le réaffecter, ce qui suppose de le rendre temporairement indisponible, comme décrit ci-après.

### Le fichier doit être déplacé ou réaffecté

Si un fichier en cours d'utilisation normale doit être déplacé ou étendu, il peut être temporairement mis hors d'utilisation. En cas de demande d'API visant à utiliser ce fichier alors qu'il est indisponible, le code anomalie MQRD\_DATA\_SET\_NOT\_AVAILABLE . est généré.

1. Utilisez la commande **RESET SMDS** pour marquer le fichier comme **ACCESS(DISABLED)**. Celui-ci est alors fermé normalement et libéré par tous les gestionnaires de files d'attente actuellement connectés.
2. Déplacez ou réallouez le fichier si nécessaire, en copiant l'ancien contenu dans le fichier nouvellement alloué, par exemple à l'aide de la commande **REPRO** d'Access Method Services (AMS).

---

<sup>1</sup> (Hits / (Hits+Misses))\* 100

Ne tentez pas de préformater le nouveau fichier avant d'y avoir copié les anciennes données, car les données copiées seraient alors ajoutées à la fin du fichier formaté.

3. Utilisez la commande **RESET SMDS** pour marquer à nouveau le fichier comme **ACCESS(ENABLED)** afin de le remettre en service.

Si la taille de l'ancien contenu est inférieure à celle du nouveau contenu, l'espace restant est automatiquement préformaté lors de l'ouverture du nouveau fichier.

Si la taille de l'ancien contenu est supérieure à celle du nouveau fichier, le gestionnaire de files d'attente doit analyser les messages de la structure de l'unité de couplage et régénérer la mappe d'espace pour vérifier qu'aucune donnée active n'a été perdue. Si une référence est trouvée à un bloc de données qui se trouve en dehors des nouvelles extensions, le fichier est marqué comme **STATUS(FAILED)** et doit être réparé en remplaçant le fichier par un fichier de taille correcte et en copiant à nouveau l'ancien fichier ou en utilisant **RECOVER CFSTRUCT** pour récupérer les messages persistants.

### L'espace de la structure d'unité de couplage est insuffisant

Si l'espace de la structure d'unité de couplage devient insuffisant et que le message IXC585E est généré, nous vous recommandons de vérifier si les règles de déchargement ont été définies de sorte à décharger la quantité maximale de données dans ces circonstances. Si tel n'est pas le cas, vous pouvez modifier les règles de déchargement à l'aide de la commande **ALTER CFSTRUCT**.

## Situations d'erreur pour les fichiers de messages partagés

Un certain nombre de problèmes ne pouvant survenir dans des circonstances normales peuvent cependant se produire en cas d'erreur.

### Impossible d'ouvrir un fichier appartenant à un gestionnaire de files d'attente

Si le gestionnaire de files d'attente qui possède un fichier de messages partagé ne peut pas l'allouer ou l'ouvrir ou si les attributs de fichier ne sont pas pris en charge, le gestionnaire de files d'attente définit une valeur de statut **SMDSCONN** appropriée de **ALLOCFAIL** ou **OPENFAIL** et la disponibilité **SMDSCONN** sur **AVAIL(ERROR)**. Il définit également la disponibilité SMDS sur **ACCESS(SUSPENDED)**. Une fois l'erreur corrigée, utilisez la commande **RESET SMDS** pour définir **ACCESS(ENABLED)** afin de déclencher une nouvelle tentative ou exécutez la commande **START SMDSCONN** sur le gestionnaire de files d'attente propriétaire.

### Impossible d'ouvrir un fichier en lecture seule

Si un gestionnaire de files d'attente ne peut pas allouer ou ouvrir un fichier de messages partagé appartenant à un autre gestionnaire de files d'attente et marqué comme **STATUS(ACTIVE)**, il suppose que cela est dû à un problème spécifique lié à sa connexion au fichier (représenté par l'objet **SMDSCONN**) plutôt qu'à un problème lié au fichier lui-même.

Il associe **SMDSCONN** à **STATUS(ALLOCFAIL)** ou **STATUS(OPENFAIL)** selon le cas et la disponibilité **SMDSCONN** à **AVAIL(ERROR)** pour empêcher toute autre tentative d'utilisation.

Si le problème peut être résolu sans affecter le statut du fichier lui-même, utilisez la commande **START SMDSCONN** pour déclencher une nouvelle tentative.

Si le problème se révèle être un problème avec le fichier lui-même, la commande **RESET SMDS** peut être utilisée pour marquer le fichier comme **STATUS(FAILED)** jusqu'à ce qu'il ait été récupéré. Une fois le fichier récupéré, l'action de retour à l'état **STATUS(ACTIVE)** entraîne la notification des autres gestionnaires de files d'attente. Si **SMDSCONN** est associé à **AVAIL(ERROR)**, il est automatiquement redéfini sur **AVAIL(NORMAL)** pour déclencher une nouvelle tentative d'ouverture du fichier.

### L'en-tête du fichier est endommagé

Si le fichier a pu être ouvert mais que le format des informations d'en-tête est incorrect, le gestionnaire de files d'attente ferme et libère le fichier, et définit le statut sur **STATUS(FAILED)**

et la disponibilité sur **ACCESS (SUSPENDED)**. Cela permet à **RECOVER CFSTRUCT** d'être utilisé pour récupérer le contenu.

Si l'erreur était liée à la présence de données résiduelles dans le fichier suite à une autre utilisation et à l'absence de préformatage du fichier par la suite, préformatez le fichier et utilisez la commande **RESET SMDS** pour définir son statut sur **STATUS (RECOVERED)**.

Sinon, le fichier doit être récupéré.

#### **Le fichier est vide de manière inattendue**

Si le gestionnaire de files d'attente ouvre un fichier marqué comme **STATUS (ACTIVE)** et détecte qu'il n'est pas initialisé ou qu'il a été préformaté mais qu'il est tout de même valide, il ferme et libère le fichier de messages partagé, puis définit le statut **STATUS (FAILED)** et la disponibilité **ACCESS (SUSPENDED)**.

#### **Le fichier contient des erreurs d'E-S permanentes**

Si un fichier contient des erreurs d'E-S permanentes après une opération **OPEN** ayant abouti, sa récupération est probablement nécessaire. Le gestionnaire de files d'attente marque le fichier comme **STATUS (FAILED)** de sorte que tous les gestionnaires de files d'attente connectés le ferment et le libèrent.

#### **Le fichier contient des erreurs d'E-S récupérables**

Si des problèmes matériels sont associés au fichier, ceux-ci peuvent entraîner des erreurs d'E-S récupérables qui ne sont pas reflétées dans le gestionnaire de files d'attente, mais qui risquent d'entraîner une dégradation importante des performances et de signaler un risque d'erreurs d'E-S permanentes dans un futur proche.

Dans ce cas, le fichier peut être mis hors ligne pour récupération à l'aide de la commande **RESET SMDS** pour le marquer comme **STATUS (FAILED)**. Le fichier est alors fermé et désaffecté par tous les gestionnaires de files d'attente. Il peut donc par exemple être déplacé vers un nouveau volume avant de redevenir disponible.

Lorsqu'un fichier devient indisponible de cette manière, la mappe d'espace n'est pas sauvegardée et l'opération de redémarrage de la connexion du gestionnaire de files d'attente devra analyser la structure d'unité de couplage pour localiser les messages du fichier et régénérer la mappe d'espace avant que le fichier ne redevienne disponible. Sinon, si le fichier de messages partagés est toujours utilisable, il peut être rendu indisponible de manière plus douce à l'aide de la commande **RESET SMDS** pour marquer le fichier **ACCESS (DISABLED)** jusqu'à ce qu'il soit prêt à être à nouveau disponible.

#### **Le contenu de fichier est incorrect**

Le gestionnaire de files d'attente ne parvient pas à détecter directement qu'un fichier contient des données incorrectes ou des données qui ne sont pas à jour, par exemple parce que le volume contenant ce fichier a dû être restauré après une sauvegarde. Toutefois, il procède à des contrôles d'intégrité permettant de réduire au minimum les risques que les programmes d'application rencontrent des données de message incorrectes.

En vue des contrôles d'intégrité, chaque bloc de message du fichier est préfixé avec une copie de l'ID entrée de l'unité de couplage correspondant, ainsi qu'avec un horodatage unique, qui est vérifié lors de la lecture du bloc de message et avant que les données de message ne soient transférées au programme utilisateur. Si le préfixe du bloc de message ne correspond pas à l'ID entrée, (et que l'entrée de l'unité de couplage n'a pas été supprimée entre-temps), le bloc de message est considéré comme endommagé et inutilisable.

Si le message endommagé était persistant, le fichier est marqué comme **STATUS (FAILED)** et le contenu de la structure doit être récupéré à l'aide de la commande **RECOVER CFSTRUCT**. Si le message endommagé était non persistant, il n'existe aucun moyen de le récupérer. Un message de diagnostic est donc généré et l'entrée du message de l'unité de couplage correspondante est supprimée.

Si aucune mappe d'espace sauvegardée n'est disponible lors de l'ouverture du fichier, celui-ci est régénéré par une recherche dans la structure d'unité de couplage des références aux données contenues dans le fichier. Au cours de cette analyse, le gestionnaire de files d'attente exécute un certain nombre d'actions :

1. Le gestionnaire de files d'attente détermine l'emplacement du message le plus récent (le cas échéant) restant actuellement dans le fichier.
2. Le gestionnaire de files d'attente lit ce message à partir du fichier pour vérifier que le préfixe du bloc correspond à l'ID entrée du message

Ces actions ont pour but de s'assurer que le gestionnaire de files d'attente détecte tous les cas où le fichier est de bas niveau et marque le fichier comme étant FAILED. Ce contrôle ne tolère toutefois pas le cas où le fichier a été restauré à partir d'une copie précédente et qu'aucun nouveau message n'y a été ajouté depuis cette restauration ou que tous les messages qui lui ont été ajoutés depuis cette copie ont été ensuite lus et supprimés.

Pour se protéger contre les données de bas niveau au cas où le fichier a été fermé normalement, le gestionnaire de files d'attente exécute un certain nombre d'actions :

1. Le gestionnaire de files d'attente sauvegarde une copie de l'horodatage de la mappe d'espace dans l'objet SMDS de Db2 lors de la fermeture normale du fichier.
2. Le gestionnaire de files d'attente vérifie ensuite si l'horodatage de la mappe d'espace est identique et détermine à quel moment le fichier est ouvert à nouveau

Si l'horodatage n'a pas la valeur souhaitée, cela peut signifier qu'une copie de bas niveau du fichier a peut-être été utilisée. Le gestionnaire de files d'attente ignore alors la mappe d'espace existante et la régénère, ce qui ne peut aboutir que si aucune donnée de message n'a été réellement perdue.

**Remarque :** Ces contrôles d'intégrité ne garantissent pas qu'un fichier de bas niveau ou endommagé soit détecté dans tous les cas théoriquement possibles. Ils ne permettent par exemple pas de détecter les cas dans lesquels le début d'un bloc de message est valide, alors que le reste des données a été en partie remplacé.

## Scénarios de récupération pour les fichiers de messages partagés

Cette section décrit les scénarios de récupération des fichiers de messages partagés.

### Récupération d'un fichier lorsqu'aucune donnée n'a été perdue

Dans certains cas, il est possible de restaurer le contenu correct d'un fichier ayant échoué sans procéder à une véritable récupération. Par exemple, lorsqu'un fichier contient des données résiduelles consécutives à une utilisation précédente et qu'il n'a pas été préformaté de nouveau. Pour résoudre le problème, il suffit de le préformater. Autre exemple : lorsqu'un fichier a été déplacé, mais qu'une erreur s'est produite lors de la copie des données. Pour résoudre le problème, il suffit de recopier correctement les données.

Dans de tels cas, le fichier corrigé peut être à nouveau disponible à l'aide de la commande **RESET SMDS** pour définir **STATUS (RECOVERED)**. Si la disponibilité est **ACCESS (SUSPENDED)**, elle redevient automatiquement **ACCESS (ENABLED)**.

Lorsque le gestionnaire de files d'attente propriétaire est averti que le fichier a été récupéré, il analyse le contenu de la structure pour reconstruire la mappe d'espace, puis remplace le statut par **STATUS (ACTIVE)**. Les autres gestionnaires de files d'attente peuvent alors recommencer à lire le fichier.

### Récupération de fichier avec TYPE(NORMAL)

Si le contenu d'un fichier a été perdu, mais que la structure applicative a été définie avec **RECOVER (YES)** et que les journaux de reprise appropriés sont disponibles, la commande **RECOVER CFSTRUCT** permet de récupérer les messages persistants stockés dans cette structure, y compris les données des messages persistants déchargés dans les fichiers de messages partagés. Cette commande restaure l'état actuel du fichier à l'aide des informations consignées par la commande

**BACKUP CFSTRUCT** et de toutes les modifications apportées aux messages persistants et consignées depuis la sauvegarde.

La commande **RECOVER CFSTRUCT** récupère toujours tous les messages persistants dans la structure d'unité de couplage avec les données de message déchargées stockées dans Db2. Pour les données déchargées stockées dans des fichiers de messages partagés, chaque fichier est sélectionné pour l'opération de récupération uniquement s'il est déjà marqué comme **STATUS (FAILED)** ou s'il est inopinément vide ou non valide lorsqu'il est ouvert par le traitement de la récupération. Il est inutile de récupérer un fichier de messages partagés marqué comme actif et qui transmet les contrôles de validation, car les données de message existantes sont déjà correctes. L'en-tête est toutefois mis à jour pour indiquer que les mappes d'espace sauvegardées devront être régénérées après la récupération.

L'opération de récupération est uniquement possible lorsque la structure a été marquée comme ayant échoué car la totalité du contenu de la structure doit être reconstruite par cette opération. Toutefois, si au moins un fichier de messages partagés a été marqué comme ayant échoué, la commande **RECOVER CFSTRUCT** va automatiquement marquer cette structure comme ayant échoué si cette action est nécessaire à la poursuite de l'opération de récupération.

La récupération peut être exécutée à partir de n'importe quel gestionnaire de files d'attente du groupe de partage de files d'attente, à condition qu'il bénéficie d'un accès en écriture aux fichiers concernés.

Seuls les messages persistants sont sauvegardés et consignés. L'opération normale de récupération les restaure donc tous, mais les messages non persistants de la structure sont perdus.

Une fois la récupération terminée, tout fichier qui a été sélectionné pour la récupération est associé automatiquement à **STATUS (RECOVERED)**, et si la disponibilité est **ACCESS (SUSPENDED)**, à **ACCESS (ENABLED)**. Le gestionnaire de files d'attente régénère la mappe d'espace pour chaque fichier en analysant les messages sur l'unité de couplage, puis marque le fichier comme **STATUS (ACTIVE)** pour qu'il puisse être réutilisé.

### Récupération de fichier avec **TYPE(PURGE)**

Pour une structure récupérable, si le contenu du fichier a été perdu, mais que la récupération n'est pas possible pour une raison quelconque, par exemple parce que les journaux de récupération ne sont pas disponibles ou que la récupération prendrait trop de temps, la commande **RECOVER CFSTRUCT** peut être utilisée avec **TYPE (PURGE)** pour rétablir l'état utilisable de la structure. La structure sera vide et tous les fichiers associés seront marqués comme **STATUS (EMPTY)**.

### Suppression de la structure applicative

Si une structure d'application non récupérable est supprimée à l'aide de la commande MVS **SETXCF FORCE** ou en raison d'un échec de la structure, lors de la prochaine connexion de la structure, le message CSQE028I est émis pour indiquer que la structure a été réinitialisée et que tous les messages existants ont été supprimés, et tous les fichiers existants sont automatiquement réinitialisés à **STATUS (EMPTY)** également. Une structure non récupérable devient à nouveau utilisable après une perte de données subie par celle-ci ou par l'un des fichiers associés.

Si une structure applicative récupérable est supprimée, elle est traitée de la même manière que si la structure avait échoué.

### Echec de la récupération d'un fichier

Si **RECOVER CFSTRUCT** ne peut pas s'exécuter pour une raison quelconque, par exemple parce qu'un fichier journal n'est plus disponible ou parce que le gestionnaire de files d'attente s'est arrêté alors que la récupération était en cours, tout fichier pour lequel la récupération a été au moins démarrée sera marqué dans l'en-tête pour indiquer que la récupération partielle a été tentée et le fichier restera à l'état **STATUS (FAILED)**.

Dans ce cas, vous pouvez réitérer la demande de récupération originale ou procéder à la récupération avec **TYPE (PURGE)** à la place, en supprimant les données existantes.

Si vous tentez de marquer le fichier comme **STATUS (RECOVERED)** sans le récupérer, à la prochaine ouverture du fichier, le gestionnaire de files d'attente verra que l'en-tête indique une récupération incomplète et le marquera à nouveau comme **STATUS (FAILED)**.

## Reprise après incident hors site

En cas de reprise après incident hors site, les messages partagés persistants peuvent être recréés en utilisant uniquement les journaux et les objets Db2 partagés contenant les définitions CFSTRUCT et les informations sur le statut de fichier de messages partagés associées.

Une fois que les tables Db2 contenant les définitions ont été configurées, il est possible de configurer la structure applicative et les fichiers de messages partagés comme étant vides. Lorsqu'un gestionnaire de files d'attente s'y connecte et détecte qu'ils sont vides de manière inattendue, il les marque comme ayant échoué, après quoi une seule commande **RECOVER CFSTRUCT** peut être utilisée pour récupérer tous les messages persistants pour toutes les structures affectées.

## **Commandes associées aux fichiers de messages partagés**

Cette rubrique décrit les commandes relatives aux fichiers de messages partagés et explique comment y accéder.

Affichez et modifiez les options **CFSTRUCT** relatives au déchargement des messages de grande taille (**OFFLOAD** et règles de déchargement) et aux fichiers de messages partagés (**DSGROUP**, **DSBLOCK**, **DSBUFS**, **DSEXPAND**) :

- [DISPLAY CFSTRUCT](#)
- [DEFINE CFSTRUCT](#)
- [ALTER CFSTRUCT](#)
- [DELETE CFSTRUCT](#)

Affichez le statut **CFSTRUCT** relatif au déchargement des messages de grande taille (**OFFLDUSE**) :

- [DISPLAY CFSTATUS](#)

Affichez et modifiez les options de remplacement des fichiers (**DSEXPAND** et **DSBUFS**) pour des gestionnaires de files d'attente donnés :

- [DISPLAY SMDS](#)
- [ALTER SMDS](#)

Affichez ou modifiez le statut et la disponibilité des fichiers dans le groupe de partage de files d'attente :

- [DISPLAY CFSTATUS TYPE\(SMDS\)](#)
- [RESET SMDS](#)

Affichez les informations relatives à l'utilisation de l'espace des fichiers de messages partagés à l'utilisation de la mémoire tampon pour un gestionnaire de files d'attente :

- [DISPLAY USAGE TYPE\(SMDS\)](#)

Affichez ou modifiez le statut et la disponibilité des connexions (**SMDSCONN**) aux fichiers à partir d'un gestionnaire de files d'attente donné :

- [DISPLAY SMDSCONN](#)
- [START SMDSCONN](#)
- [STOP SMDSCONN](#)

Sauvegardez et restaurez les messages partagés, notamment les données des messages de grande taille dans SMDS, le cas échéant :

- [BACKUP CFSTRUCT](#)
- [RECOVER CFSTRUCT](#)

## **Avantages liés à l'utilisation des files d'attente partagées**

Les files d'attente partagées permettent aux applications IBM MQ d'être évolutives, hautement disponibles et d'implémenter l'équilibrage de charge.

## Avantages liés aux files d'attente partagées

L'architecture en files d'attente partagées, dans laquelle des serveurs clonés extraient des tâches d'une file d'attente partagée unique, se caractérise par les propriétés suivantes :

- Vous pouvez la faire évoluer en ajoutant de nouvelles instances de l'application serveur ou même une nouvelle image z/OS avec un gestionnaire de files d'attente (dans le groupe de partage de files d'attente) et une copie de l'application.
- Elle est hautement disponible.
- Elle procède naturellement à l'*extraction* de l'équilibrage de charge en fonction de la capacité de traitement disponible de chaque gestionnaire de files d'attente du groupe de partage de files d'attente.

## Utilisation des files d'attente partagée en vue de la haute disponibilité

Les exemples suivants illustrent les utilisations possibles d'une file d'attente partagée en vue d'améliorer la disponibilité de l'application.

Supposons un scénario IBM MQ dans lequel des applications client s'exécutant dans le réseau souhaitent effectuer des demandes d'applications serveur qui s'exécutent sur z/OS. L'application client construit un message de demande et le place dans une file d'attente des demandes. Le client attend alors une réponse du serveur, qui est envoyée à la file d'attente de réponse nommée dans le descripteur du message de demande.

IBM MQ gère le transport du message de demande entre la machine client et la file d'entrée du serveur sur z/OS, ainsi que le transport de la réponse renvoyée au client par le serveur. Lorsque la file d'entrée du serveur est définie en tant que file d'attente partagée, les messages placés dans la file d'attente peuvent être extraits par n'importe quel gestionnaire de files d'attente du groupe de partage. Cela signifie que vous pouvez configurer un gestionnaire de files d'attente dans chaque image z/OS du sysplex, et en les connectant tous au même groupe de partage de files d'attente, chacun d'entre eux peut accéder aux messages dans la file d'attente d'entrée du serveur.

Les messages de la file d'entrée du serveur sont toujours disponibles, même si l'un des gestionnaires de files d'attente s'arrête de manière anormale ou si vous devez l'arrêter pour des raisons d'administration. Vous pouvez mettre l'intégralité d'une image z/OS hors ligne, et les messages restent disponibles.

Pour tirer parti de la disponibilité des messages sur une file d'attente partagée, exécutez une instance de l'application serveur dans chaque image z/OS du sysplex afin de bénéficier d'une plus grande capacité et disponibilité de cette application, comme indiqué dans la [Figure 61](#), à la [page 199](#).

Une instance de l'application serveur extrait un message de demande de la file d'attente partagée et, selon son contenu, le traite et génère un résultat qui est renvoyé au client sous la forme d'un message IBM MQ. Le message de réponse est destiné à la file d'attente de réponse et au gestionnaire de files d'attente de réponse nommés dans le descripteur du message de demande.

Vous pouvez configurer le chemin de retour à l'aide d'un certain nombre d'options. Pour plus d'informations sur ces options, voir [«Mise en file d'attente répartie et groupes de partage de files d'attente»](#), à la [page 220](#).

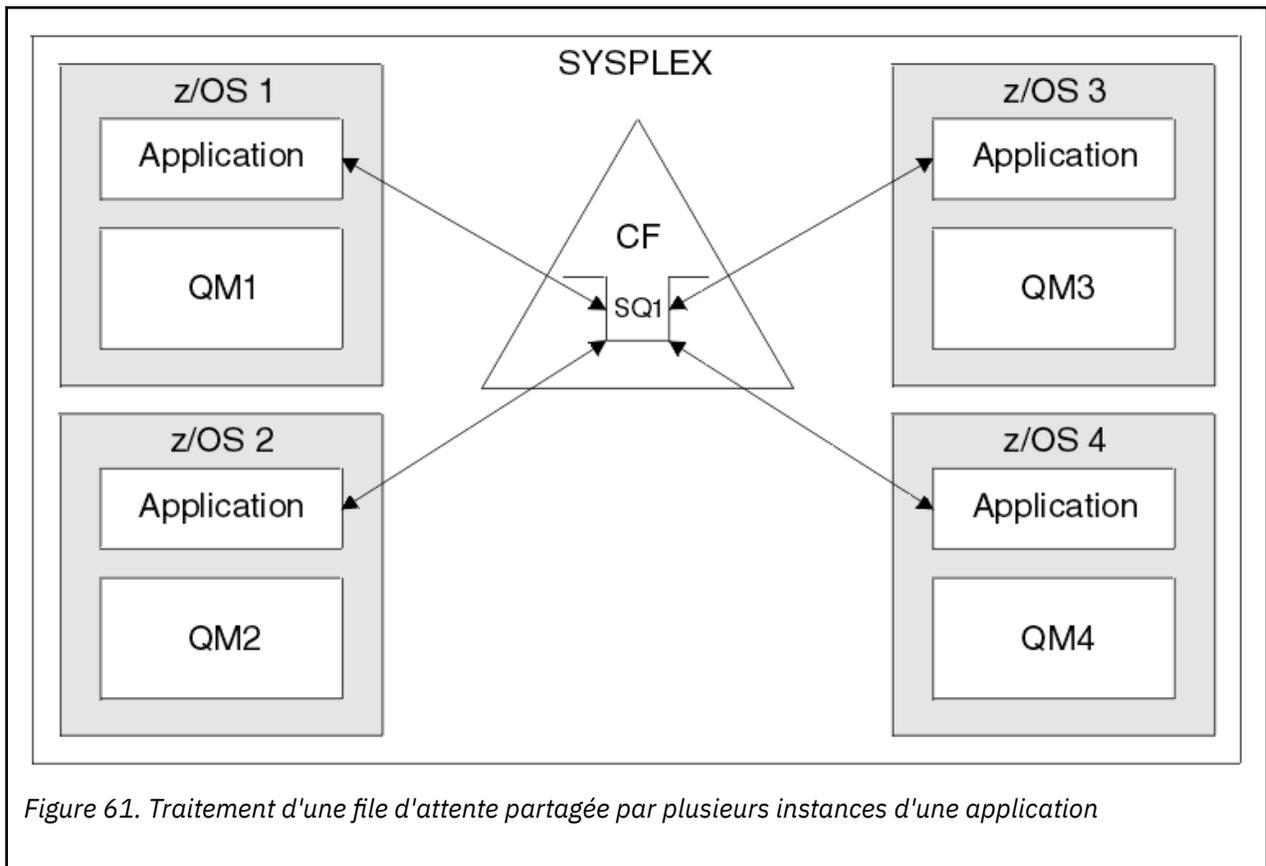


Figure 61. Traitement d'une file d'attente partagée par plusieurs instances d'une application

## Reprise homologue

Pour améliorer la disponibilité des messages dans un groupe de partage de files d'attente, IBM MQ détecte si un autre gestionnaire de files d'attente du groupe se déconnecte de manière anormale de l'unité de couplage et termine les unités d'oeuvre toujours en attente pour ce gestionnaire de files d'attente lorsque c'est possible. Cette fonction s'appelle *reprise homologue*.

Supposons qu'un gestionnaire de files d'attente s'arrête de manière anormale au moment où une application a extrait un message de demande d'une file d'attente du point de synchronisation, mais n'a pas encore placé le message de réponse ni validé l'unité d'oeuvre. Un autre gestionnaire de files d'attente du groupe de partage détecte l'incident et annule les unités en cours exécutées sur le gestionnaire de files d'attente défaillant. Cela signifie que le message de demande est replacé dans la file d'attente des demandes en vue de son traitement par l'une des autres instances de serveur à traiter sans qu'il soit nécessaire d'attendre le redémarrage du gestionnaire de files d'attente défaillant.

Si IBM MQ ne parvient pas à résoudre automatiquement une unité d'oeuvre, vous pouvez résoudre manuellement la partie partagée pour activer un autre gestionnaire de files d'attente du groupe de partage de files d'attente et poursuivre le traitement de cette unité.

## ► z/OS **Utilisation de la mémoire de classe d'archivage avec des files d'attente partagées**

L'utilisation de la mémoire de classe d'archivage, combinée à des files d'attente partagées IBM MQ for z/OS, peut présenter des avantages.

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

Les machines z13, zEC12 et zBC12 autorisent l'installation des cartes Flash Express. Ces dernières contiennent des unités SSD flash. Après l'installation, le stockage flash fourni par les cartes peut être alloué à une ou plusieurs partitions logiques ; il est désigné sous le terme mémoire de classe d'archivage (SCM).

La mémoire de classe d'archivage (SCM) se situe entre la mémoire réelle et le périphérique de stockage à accès direct (DASD) en termes de délai d'attente des E-S et de coût. Cette mémoire ne comportant pas d'éléments mobiles, les délais d'attente des E-S sont bien inférieurs à ceux des périphériques de stockage à accès direct.

La mémoire de classe d'archivage est également bien moins coûteuse que la mémoire réelle. Il est ainsi possible d'installer un grand volume de stockage pour un coût relativement faible. Par exemple, une paire de cartes Flash Express peut contenir 1424 Go de mémoire utilisable.

Ces caractéristiques signifient que la mémoire de classe d'archivage est utile lorsqu'une grosse quantité de données doit être retirée de la mémoire réelle en un court laps de temps car les données peuvent être enregistrées dans la mémoire de classe d'archivage plus rapidement que sur des périphériques de stockage à accès direct (DASD). Cet élément peut être particulièrement utile lors de l'utilisation d'une structure de liste d'unité de couplage (CF) contenant des files d'attente partagées IBM MQ.

## Raisons pour lesquelles les structures de liste sont saturées

Lorsqu'une structure d'unité de couplage est définie, elle est configurée avec un attribut SIZE qui décrit la taille maximale de la structure. Les structures d'unité de couplage résidant de manière permanente dans la mémoire réelle, la somme des attributs SIZE des structures définies dans une unité de couplage doit être inférieure à la quantité de mémoire réelle allouée à l'unité de couplage.

Il existe ainsi une pression constante pour conserver la valeur SIZE d'une structure donnée à la valeur maximale possible afin que des structures supplémentaires puissent tenir dans l'unité de couplage. Toutefois, s'assurer que les structures sont suffisamment grandes pour remplir leur objectif peut engendrer une pression contradictoire car rendre une structure trop petite signifie qu'elle risque d'être saturée ce qui perturbe les applications ou les sous-systèmes qui en font usage.

Il est crucial de définir avec précision la taille d'une structure en fonction de son usage attendu. Toutefois, cette tâche est difficile à exécuter car les charges de travail peuvent varier au fil du temps et prévoir leur fluctuation n'est pas facile.

Les files d'attente partagées IBM MQ utilisent les structures de liste d'unité de couplage pour stocker les messages. IBM MQ appelle des structures d'unité de couplage qui contiennent des messages et des structures d'application.

Les structures d'application sont référencées à l'aide des informations stockées dans des objets CFSTRUCT IBM MQ. Lorsqu'un message d'une taille inférieure à 63 ko est inséré dans une file d'attente partagée, il est stocké complètement dans une structure d'application sous la forme d'une entrée de liste unique et aucun ou plusieurs éléments de liste.

Comme les files d'attente partagées IBM MQ utilisent des structures de liste, les pressions décrites affectent également les files d'attente partagées. Dans ce cas, le nombre maximal de messages pouvant être stockés dans une file d'attente partagée est une fonction :

- de la taille des messages dans la file d'attente,
- de la taille maximale de la structure,
- du nombre des entrées et des éléments disponibles dans la structure.

Etant donné que 512 files d'attente partagées peuvent utiliser la même structure et entrent en compétition pour les entrées et les éléments, cela complique encore plus la situation.

Les files d'attente IBM MQ sont utilisées pour le transfert des données entre les applications. Il est donc habituel qu'une application place des messages dans une file d'attente lorsque l'application partenaire qui doit recevoir ces messages n'est pas en cours d'exécution.

Lorsque cela arrive, le nombre de messages dans la file d'attente augmente au fil du temps jusqu'à ce qu'une ou plusieurs des situations ci-dessous se produisent.

- L'application d'insertion arrête d'insérer des messages.
- L'application d'extraction des messages commence à extraire des messages.
- Les messages qui se trouvent dans la file d'attente commencent à arriver à expiration et sont supprimés de la file d'attente.
- La file d'attente atteint sa taille maximale, auquel cas le code raison MQRC\_Q\_FULL est renvoyé à l'application d'insertion.
- La structure contenant la file d'attente partagée atteint sa taille maximale ou l'unité de couplage contenant la structure manque de mémoire disponible. Dans l'un ou l'autre cas, le code raison MQRC\_STORAGE\_MEDIUM\_FULL est renvoyé à l'application d'insertion.

Dans les trois derniers cas, la file d'attente est saturée. A ce stade, l'application d'insertion n'a plus d'emplacement où insérer les messages. Elle peut résoudre le problème en utilisant une ou plusieurs solutions suivantes :

- En effectuant des tentatives répétées d'insertion du message en respectant un certain délai entre chacune d'elles.
- En insérant les messages à un emplacement différent, comme dans une base de données ou un fichier. Les messages sont accessibles ultérieurement et insérés normalement dans la file d'attente.
- En supprimant le message s'il n'est pas persistant.

Toutefois, pour certaines classes d'application, par exemple celles qui comportent un volume important de messages entrants ou qui ne disposent pas d'un accès au système de fichiers, ces solutions ne sont pas pratiques. Il convient de s'assurer en tout premier lieu que les files d'attente ne seront jamais ou très rarement saturées, surtout dans le cas des files d'attente partagées.

## Service SMDS et règles de déchargement

Les règles de déchargement introduites dans IBM WebSphere MQ 7.1 permettent de réduire les probabilités de saturation d'une structure d'application.

Chaque structure d'application est associée à trois règles définies à l'aide de trois paires de mots clés :

- OFFLD1SZ et OFFLD1TH
- OFFLD2SZ et OFFLD2TH
- OFFLD3SZ et OFFLD3TH

Chaque règle indique les conditions qui doivent être remplies pour que les données de message soit déchargées dans le mécanisme de stockage qui est associé à la structure d'application. Deux types de mécanismes de stockage sont actuellement disponibles :

- Db2
- Groupe de fichiers linéaires VSAM (Virtual Storage Access Method) appelé par IBM MQ fichier de messages partagé (SMDS).

L'exemple suivant présente la commande MQSC permettant de créer une structure d'application nommée LIST1, à l'aide de la commande DEFINE CFSTRUCT.

Cette structure applique les règles de déchargement par défaut et utilise le fichier de messages partagé comme mécanisme de déchargement. Cela signifie que lorsque la structure est pleine à 70 % (OFFLD1TH), tous les messages d'une taille de 32 ko ou supérieure (OFFLD1SZ) sont déchargés dans le fichier de messages partagé.

De même, lorsque la structure est pleine à 80 % (OFFLD2TH), tous les messages d'une taille de 4 ko ou supérieure (OFFLD2SZ) sont déchargés. Lorsque la structure est pleine à 90 % (OFFLD3TH), tous les messages (OFFLD3SZ) sont déchargés.

```
DEFINE CFSTRUCT(LIST1)
```

```
CFLEVEL(5)
OFFLOAD(SMDS)
OFFLD1SZ(32K) OFFLD1TH(70)
OFFLD2SZ(4K) OFFLD2TH(80)
OFFLD3SZ(0K) OFFLD3TH(90)
```

Tous les messages déchargés sont stockés sur le support de déchargement et un pointeur vers le message est stocké dans la structure. Si les règles de déchargement réduisent les risques de saturation de la structure en plaçant moins de données dans cette dernière lorsque l'espace commence à manquer, des données sont toujours enregistrées dans la structure pour chaque message. C'est-à-dire le pointeur vers le message déchargé.

De plus, les règles de déchargement offrent un avantage en terme de performance. L'enregistrement d'un message dans une structure est relativement rapide et largement dominé par la durée nécessaire à l'envoi de la demande d'enregistrement dans l'unité de couplage. L'enregistrement réel dans la structure est rapide, et s'effectue à la vitesse de la mémoire réelle.

L'enregistrement d'un message dans un fichier de messages partagé est bien plus lent car cela comprend l'enregistrement dans la structure du pointeur de message et l'enregistrement des données de message dans le fichier de messages partagé. Cette deuxième opération d'enregistrement s'effectue à la vitesse du périphérique de stockage à accès direct et l'ajout éventuel d'un délai d'attente. Si Db2 est utilisé comme mécanisme de déchargement, le coût de performance est bien supérieur.

### Comment la mémoire de classe d'archivage fonctionne avec IBM MQ for z/OS

Présentation de l'utilisation de la mémoire de classe d'archivage (SCM) avec des files d'attente partagées IBM MQ for z/OS.

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

Il est possible d'allouer une mémoire de classe d'archivage à une unité de couplage (CF) au niveau CFLEVEL 19, ou supérieur. Les structures définies dans cette unité de couplage peuvent être configurées pour utiliser la mémoire de classe d'archivage dans le but de réduire les risques de saturation des structures (condition de saturation de la structure). Lorsqu'une structure configurée pour utiliser la mémoire de classe d'archivage se remplit au-dessus d'un point déterminé par le système, l'unité de couplage commence à transférer les données de la structure vers la mémoire de classe d'archivage, ce qui libère de l'espace dans la structure pour de nouvelles données.

**Remarque :** La mémoire de classe d'archivage pouvant elle-même être saturée, l'allocation de celle-ci à une structure réduit les risques de saturation de la structure, tout en ne les éliminant pas entièrement.

Une structure est configurée pour utiliser la mémoire de classe d'archivage en spécifiant les mots clés **SCMALGORITHM** et **SCMAXSIZE** dans la règle CFRM (Coupling Facility Resource Manager) qui contient la définition de cette structure.

Notez qu'une fois ces mots clés spécifiés et la règle CFRM appliquée, la structure doit être régénérée ou désallouée afin qu'ils puissent prendre effet.

## Mot clé **SCMALGORITHM**

La vitesse d'entrée-sortie de la mémoire de classe d'archivage étant plus lente que celle de la mémoire réelle, l'unité de couplage utilise un algorithme qui est adapté à l'utilisation attendue de la structure afin de réduire l'impact de l'écriture ou de la lecture dans la mémoire de classe d'archivage.

L'algorithme est configuré par le mot clé **SCMALGORITHM** dans la règle CFRM pour la structure en utilisant la valeur *KEYPRIORITY1*. Sachez que vous ne devez utiliser la valeur *KEYPRIORITY1* qu'avec les structures de liste utilisées par les files d'attente partagées IBM MQ.

L'algorithme *KEYPRIORITY1* fonctionne en partant du principe que la plupart des applications extrairont les messages d'une file d'attente partagée par ordre de priorité, c'est-à-dire que lorsqu'une application extrait un message, elle obtient le message le plus ancien ayant la priorité la plus élevée.

Lorsqu'une structure commence à se remplir au-dessus du seuil de 90 % défini par le système, l'unité de couplage commence à migrer de manière asynchrone les messages qui ont le moins de chance d'être extraits. Il s'agit des messages ayant la priorité la plus faible qui ont été insérés récemment dans la file d'attente.

Cette migration asynchrone des messages de la structure vers la mémoire de classe d'archivage est appelée "pré-transfert".

Le pré-transfert réduit le coût de performance lié à l'utilisation de la mémoire de classe d'archivage car il réduit les risques de blocage d'une application lorsque des entrées-sorties synchrones sont dirigées vers la mémoire de classe d'archivage.

Outre le pré-transfert, l'algorithme *KEYPRIORITY1* renvoie également de manière asynchrone les messages depuis la mémoire de classe d'archivage et vers la structure lorsqu'un espace suffisant est disponible. Pour l'algorithme *KEYPRIORITY1*, cela signifie que la structure est plus ou moins pleine à 70 %.

Le transfert des messages de la mémoire de classe d'archivage SCM vers la structure est appelé "pré-extraction".

La pré-extraction réduit les chances qu'une application tente d'extraire un messages qui a été pré-transféré vers la mémoire de classe d'archivage et de devoir attendre que l'unité de couplage renvoie de manière asynchrone les messages dans la structure.

## Mot clé **SCMMAXSIZE**

Le mot clé **SCMMAXSIZE** définit la quantité maximale de mémoire de classe d'archivage qui peut être utilisée par une structure. La mémoire de classe d'archivage étant allouée à la structure par l'unité de couplage lorsque cela est nécessaire, il est possible de spécifier un **SCMMAXSIZE** qui est supérieur à la quantité totale de mémoire de classe d'archivage disponible. C'est ce qu'on appelle la "sur-affectation".

**Important :** Ne sur-affectez jamais la mémoire de classe d'archivage. En effet, les applications qui reposent sur cette mémoire n'obtiendraient pas le comportement attendu. Par exemple, les applications IBM MQ qui utilisent des files d'attente partagées peuvent recevoir des codes raison MQRC\_STORAGE\_MEDIUM\_FULL inattendus.

L'unité de couplage utilise des structures de données pour suivre son utilisation de la mémoire de classe d'archivage. Ces structures de données se trouvent dans la mémoire réelle qui est allouée à l'unité de couplage, ce qui a pour résultat de réduire la quantité de mémoire réelle pouvant être utilisée par les structures. La mémoire utilisée par ces structures de données est appelée "espace augmenté".

Lorsqu'une structure est configurée avec la mémoire de classe d'archivage, une petite quantité de mémoire réelle est allouée depuis l'unité de couplage vers la structure, ce qui est appelé espace augmenté fixe. Ceci est alloué même si la structure n'utilise jamais vraiment la mémoire de classe d'archivage. Comme les données de la structure sont stockées dans la mémoire de classe d'archivage, un espace augmenté dynamique supplémentaire sera alloué depuis la mémoire réelle de secours dans l'unité de couplage.

Lorsque les données sont supprimées de la mémoire de classe d'archivage, l'espace augmenté dynamique est renvoyé à l'unité de couplage. L'espace augmenté, qu'il soit fixe ou dynamique, n'est jamais prélevé dans la mémoire réelle qui est allouée à une structure.

Outre l'espace augmenté, lorsqu'une structure est configurée pour utiliser la mémoire de classe d'archivage, la quantité de mémoire de contrôle utilisée par cette structure augmente. Cela signifie qu'une structure de liste configurée avec la mémoire de classe d'archivage peut contenir moins d'entrées et d'éléments qu'une structure de même taille sans mémoire de classe d'archivage configurée.

Pour comprendre l'impact de la mémoire de classe d'archivage sur des structures nouvelles ou existantes, utilisez l'outil [CFSizer](#).

Un point final important à noter est qu'une fois que les données ont été transférées depuis la structure vers la mémoire de classe d'archivage et que l'espace augmenté dynamique a été utilisé, la structure ne peut pas être modifiée manuellement ou automatiquement.

En d'autres termes, la quantité de mémoire allouée à la structure ne peut être ni augmentée ni réduite, le rapport entrées/éléments utilisé par la structure ne peut pas être modifié, etc. Pour qu'elle soit à nouveau modifiable, la structure ne doit plus contenir de données stockées dans la mémoire de classe d'archivage et elle ne doit plus utiliser l'espace augmenté dynamique.

 *Pour quelle raison utiliser la mémoire de classe d'archivage (SCM)*

L'archivage d'urgence et le gain de performances sont les deux cas d'utilisation de la mémoire de classe d'archivage avec IBM MQ for z/OS.

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

Cette section présente la théorie sur laquelle sont fondés les deux scénarios possibles. Pour plus de détails sur la configuration des scénarios, voir :

- [«Archivage d'urgence - configuration de base»](#), à la page 208
- [«Gain de performances - configuration de base»](#), à la page 214

**Important :** L'utilisation de la mémoire de classe d'archivage avec des structures CF n'est pas liée à une version spécifique d'IBM MQ. Toutefois, le scénario d'archivage d'urgence fonctionne uniquement avec IBM WebSphere MQ 7.1 et ultérieure car il requiert un fichier de messages partagé et les règles de déchargement.

## Archivage d'urgence

Le fichier de messages partagé et le déchargement des messages peuvent être utilisés conjointement avec la mémoire de classe d'archivage pour réduire les risques de renvoi d'un code raison MQRC\_STORAGE\_MEDIUM\_FULL à l'application IBM MQ en cas de longue indisponibilité.

### Présentation

Une file d'attente partagée unique est configurée dans une structure d'application. L'application d'insertion place des messages dans la file d'attente partagée et l'application d'extraction extrait les messages de la file d'attente partagée.

Dans le cadre de l'exécution normale, la taille de la file d'attente est sensée être proche de zéro, mais une exigence métier indique que le système doit pouvoir tolérer une indisponibilité de deux heures de l'application d'extraction. Cela signifie que la file d'attente partagée doit être capable de contenir deux heures de messages provenant de l'application d'insertion.

Actuellement, ce processus est réalisé à l'aide des règles de déchargement par défaut et du fichier de messages partagé, afin que la taille de la structure soit réduite au minimum, tout en réduisant le coût de performance qui est associé au déchargement.

Le taux des messages envoyés à la file d'attente partagée est sensé doubler à court et à moyen terme. Bien que l'exigence selon laquelle le système doit pouvoir tolérer une indisponibilité de deux heures soit toujours valable, la quantité de mémoire réelle disponible dans la structure d'unité de couplage n'est pas suffisante pour doubler la taille de la structure.

Comme l'unité de couplage (CF) qui contient la structure d'application se trouve sur une machine zEC12, il est possible d'associer une mémoire de classe d'archivage à la structure pour stocker suffisamment de messages afin qu'une indisponibilité de deux heures puisse être tolérée.

Envisagez ce qui se passe au cours d'une certaine période de temps :

1. Initialement, le système est à l'état stabilisé. L'application d'insertion et l'application d'extraction s'exécutent normalement et la taille de la file d'attente est de zéro ou proche de zéro. Le résultat est que la structure d'application est largement vide.
2. A un certain moment, l'application d'extraction subit une défaillance et s'arrête. L'application d'insertion continue à placer des messages dans la file d'attente et la structure d'application commence à se remplir.
3. Lorsque la structure est pleine à 70 %, les conditions de la première règle de déchargement sont remplies et tous les messages d'une taille supérieure ou égale à 32 ko sont déchargés dans le fichier de messages partagé.

Pour une présentation des règles de déchargement, voir [«Service SMDS et règles de déchargement»](#), à la page 201.

4. Comme les messages continuent à être placés dans la file d'attente partagée, la structure continue à se remplir (en raison du stockage des données de message dans la structure ou du stockage des pointeurs vers les messages déchargés dans la structure).

Lorsque la structure est pleine à 80 %, la deuxième règle de déchargement commence à s'appliquer et les messages d'une taille de 4 ko ou supérieure sont déchargés dans le fichier de messages partagé.

5. Lorsque la structure est pleine à plus de 90 %, tous les messages sont déchargés dans le fichier de messages partagé et seuls les pointeurs des messages sont placés dans la structure.

A ce moment, l'algorithme de pré-transfert commence à s'exécuter et à transférer les données depuis la structure vers la mémoire de classe d'archivage. En supposant que tous les messages dans la file d'attente ont la même priorité, les messages les plus récents sont pré-transférés.

Comme tous les messages sont maintenant déchargés dans le fichier de messages partagé, les données transférées dans la mémoire de classe d'archivage ne correspondent pas aux données réelles des messages mais plutôt aux pointeurs vers les messages contenus dans le fichier de messages partagé.

Ainsi, le nombre de messages pouvant être stockés dans la structure, la mémoire de classe d'archivage et le fichier de messages partagé associé à la structure est très élevé.

**Performance :** Au cours de cette période d'indisponibilité, l'application d'insertion peut subir une certaine baisse de performances dû à l'enregistrement dans le fichier de messages partagé. Dans ce cas, l'utilisation de la mémoire de classe d'archivage ne doit pas être un facteur de limitation en terme de performances pour l'application d'insertion. La mémoire de classe d'archivage fournit un espace supplémentaire permettant d'éviter la saturation de la structure.

6. Enfin, l'application d'extraction redevient disponible et l'indisponibilité est terminée.

Toutefois, la mémoire de classe d'archivage est encore utilisée par la structure. L'application d'extraction commence à lire les messages à partir de la file d'attente, en extrayant d'abord les messages les plus anciens ayant la priorité la plus élevée.

Ces messages ayant été enregistrés avant que la structure ne soit saturée, ils proviennent tous de la portion de mémoire réelle de la structure.

7. Comme la structure commence à se vider, elle passe sous le seuil où le pré-transfert est actif et le pré-transfert s'arrête.
8. L'utilisation de la structure baisse jusqu'au point où les règles de déchargement prennent effet. Les messages ne sont déchargés dans le fichier de messages partagés que si leur taille dépasse 63 ko.

A peu près à ce moment, l'algorithme de pré-extraction commence à transférer les données de la mémoire de classe d'archivage vers la structure. Comme l'application d'extraction extrait les messages de la file d'attente dans l'ordre attendu par les algorithmes SCM, les messages sont transférés avant que l'application d'extraction n'en ait besoin.

Ainsi, l'application d'extraction n'a pas besoin d'attendre que les messages soient transférés de manière synchrone depuis la mémoire de classe d'archivage.

9. Comme l'application d'extraction continue à parcourir la file d'attente, elle commence à extraire des messages qui ont été déchargés dans le fichier de messages partagé.
10. Le fichier repasse enfin à un état stabilisé. Aucun message n'est stocké dans la mémoire de classe d'archivage ni dans le fichier de messages partagé et la taille de la file d'attente est proche de zéro.

## Performances améliorées

Ce scénario décrit l'utilisation de la mémoire de classe d'archivage pour augmenter le nombre de messages pouvant être stockés dans une file d'attente partagée sans subir le coût en terme de performance que représente l'utilisation du fichier de messages partagé.

### Description

Pour ce scénario, une application d'insertion et une application d'extraction communiquent par le biais d'une file d'attente partagée qui est stockée dans la structure d'application.

L'application d'insertion a tendance à s'exécuter par rafales lorsqu'elle insère un grand nombre de messages en une court laps de temps. Puis, pendant une longue période, elle ne produit plus du tout de messages.

L'application d'extraction traite chaque message de manière séquentielle et effectue un traitement complexe sur chacun d'eux. Ainsi, la plupart du temps la taille de la file d'attente est de zéro, sauf lorsque l'application d'insertion commence à s'exécuter. Elle commence à augmenter lorsque les messages sont insérés plus vite qu'ils ne sont extraits.

La taille de la file d'attente augmente jusqu'à ce que l'application d'insertion s'arrête et l'application d'extraction dispose d'un temps suffisant pour traiter tous les messages figurant dans la file d'attente.

### Remarques :

1. Dans ce scénario, le facteur clé est la performance. Les messages envoyés à la file d'attente sont toujours d'une taille inférieure à 63 ko et ne doivent donc jamais être déchargés dans le fichier de messages partagé.
2. La structure d'application a été dimensionnée afin d'être suffisamment grande pour contenir tous les messages qui y sont placés par l'application d'insertion en une seule "rafale".
3. Les règles de déchargement doivent être toutes désactivées afin que les messages ne soient pas déchargés dans le fichier de messages partagé même si la structure commence à se remplir. Cela est dû au fait que les coûts en terme de performance qui sont associés à l'enregistrement des messages et leur lecture dans le fichier de messages partagé sont jugés inacceptables.

Au fil du temps, le nombre de messages envoyés par l'application d'insertion en une rafale doit augmenter de plusieurs ordres de grandeur. Comme l'application d'extraction doit traiter chaque message séquentiellement, le nombre de messages dans la file d'attente augmente jusqu'à saturation de la structure.

A ce stade, l'application d'insertion reçoit un code raison (MQRC\_STORAGE\_MEDIUM\_FULL) lors de l'insertion d'un message et l'opération échoue. Elle peut tolérer de courtes périodes où l'insertion de messages dans la file d'attente est impossible. Si cette période est trop longue, l'application s'arrête.

Si vous n'avez ni le temps ni les compétences pour réécrire l'application d'insertion ou l'application d'extraction, trois solutions au problème sont possibles :

1. Augmentez la taille de la structure d'application.
2. Ajoutez des règles de déchargement à la structure d'application afin que les messages soient déchargés dans le fichier de messages partagé lorsque la file d'attente commence à être saturée.
3. Associez la mémoire de classe d'archivage à la structure.

La première solution est rapide à mettre en oeuvre, mais la mémoire réelle disponible dans l'unité de couplage n'est pas suffisante.

La deuxième solution est également rapide à mettre en oeuvre, mais l'impact sur les performances du déchargement dans le fichier de messages partagé est considéré comme trop important pour utiliser cette option.

La troisième solution, qui associe la mémoire de classe d'archivage à la structure, fournit un équilibre acceptable entre le coût et les performances.

L'association de la mémoire de classe d'archivage à une structure entraîne une utilisation plus élevée de la mémoire réelle dans l'unité de couplage en raison de la quantité de mémoire supérieure utilisée par les opérations d'extraction. Toutefois, la quantité réelle de mémoire réelle reste inférieure à celle utilisée dans la première option.

Un autre élément à prendre en compte est le coût de la mémoire de classe d'archivage. Ce coût est cependant bien moins élevé que celui de la mémoire réelle. Ces facteurs combinés rendent la troisième option moins coûteuse que la première.

Bien que la troisième option puisse potentiellement ne pas fonctionner aussi bien que la première option, les algorithmes de pré-extraction et de pré-transfert utilisés par l'unité de couplage peuvent se combiner pour rendre les différences de performances acceptables, voire négligeables.

Les performances peuvent être bien meilleures qu'en utilisant le fichier de messages partagé pour décharger les messages.

Envisagez ce qui se passe au cours d'une certaine période de temps :

1. Initialement, l'application d'extraction est active et attend les messages à distribuer à la file d'attente partagée. L'application d'insertion n'est pas active et la file d'attente partagée est vide.
2. A un certain moment, l'application d'insertion devient active et commence à insérer un grand nombre de messages dans la file d'attente partagée. L'application d'extraction commence à extraire les messages, mais la taille de la file d'attente commence à augmenter car l'application d'extraction est plus lente que l'application d'insertion.

La structure d'application commence donc à se remplir.

3. Au fur et à mesure que le temps passe, l'application d'insertion est toujours active. La structure d'application est remplie à environ 90 %.

C'est à ce moment que l'algorithme de pré-transfert SCM commence à transférer les messages depuis la structure vers la mémoire de classe d'archivage, ce qui libère de l'espace dans la structure.

Comme l'application d'extraction extrait d'abord de la file d'attente les messages les plus anciens ayant la priorité la plus élevée, elle extrait toujours les messages de la structure et n'a pas besoin d'attendre que des messages soient transférés de manière synchrone de la mémoire de classe d'archivage vers la structure.

4. L'application d'insertion est toujours active et insère des messages dans la file d'attente partagée. Cependant, l'application ne reçoit jamais le code raison MQRC\_STORAGE\_MEDIUM\_FULL car l'espace existant dans la mémoire de classe d'archivage est suffisant pour stocker tous les messages qui ne tiennent pas dans la structure.
5. Finalement, l'application d'insertion s'arrête car elle n'a pas plus de messages à insérer.

L'algorithme de pré-transfert s'arrête car la structure passe sous le seuil des 90 % d'utilisation et l'application d'extraction continue à traiter les messages dans la file d'attente.

6. Comme l'application d'extraction commence à libérer de l'espace dans la structure, l'algorithme de pré-extraction commence à transférer les messages de la mémoire de classe d'archivage vers la structure.

Comme l'application d'extraction traite les messages dans l'ordre attendu par l'algorithme de pré-extraction, elle n'est jamais bloquée en attendant que les données de message soient transférées de manière synchrone de la mémoire de classe d'archivage vers la structure.

7. Finalement, l'application d'extraction traite tous les messages dans la file d'attente partagée et attend que le message suivant soit disponible. La structure et la mémoire de classe d'archivage sont vides de messages.

Comment configurer un scénario de base pour l'archivage d'urgence dans IBM MQ.

## Pourquoi et quand exécuter cette tâche

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

Le fichier de messages partagé et le déchargement des messages peuvent être utilisés conjointement avec la mémoire de classe d'archivage pour réduire les risques de renvoi d'un code raison MQRC\_STORAGE\_MEDIUM\_FULL à l'application IBM MQ en cas de longue indisponibilité.

Par exemple, votre entreprise possède une application qui insère des messages dans la file d'attente et une application qui extrait les messages de la file d'attente. Dans le cadre de l'exécution normale, vous vous attendez à ce que la taille de la file d'attente soit proche de zéro, mais une exigence métier indique que le système doit pouvoir tolérer une indisponibilité de deux heures de la part de l'application qui extrait les messages.

Cela signifie que la file d'attente partagée utilisée doit être capable de contenir deux heures de messages provenant de l'application d'insertion. Cela est réalisable actuellement à l'aide des règles de déchargement par défaut et du fichier de messages partagé.

Vous vous attendez à ce que le taux des messages envoyés à la file d'attente partagée double à court et à moyen terme. Bien que l'exigence selon laquelle le système doit pouvoir tolérer une indisponibilité de deux heures soit toujours valable, la quantité de mémoire réelle disponible dans la structure d'unité de couplage n'est pas suffisante pour doubler la taille de la structure. Comme l'unité de couplage qui contient la structure d'application se trouve sur une machine zEC12, vous avez la possibilité d'associer une mémoire de classe d'archivage à la structure pour stocker suffisamment de messages afin qu'une indisponibilité de deux heures puisse être tolérée.

Ce scénario initial utilise les éléments suivants :

- Un groupe de partage de files d'attente, IBM1, qui contient un gestionnaire de files d'attente unique, CSQ3. Outre la structure d'administration, le groupe de partage de files d'attente a défini une structure d'application unique, SCEN1.
- L'unité de couplage (CF) CF01, dans laquelle la structure d'application SCEN1 est stockée en tant que structure IBM1SCEN1. Cette structure a la taille maximale de 1 Go.
- Une file d'attente partagée unique, SCEN1.Q, utilisée par la structure d'application.

Cette configuration est illustrée dans la [Figure 62](#), à la page 209.

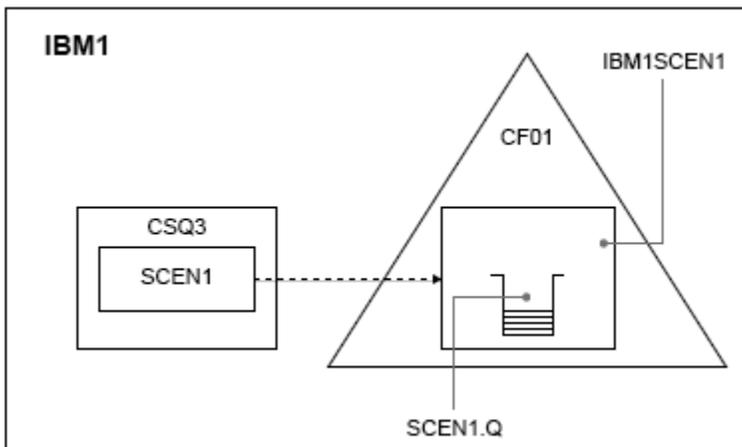


Figure 62. Configuration de base

Supposons également que le gestionnaire de files d'attente CSQ3 est déjà le seul membre du groupe de partage de files d'attente IBM1.

Vous devez ajouter la définition pour la structure IBM1SCEN1 à la règle CFRM (Coupling Facility Resource Manager). Dans un souci de simplicité, la structure est définie de manière à pouvoir être créée dans une seule unité de couplage, CF01, en spécifiant PREFLIST (CF01).



**Avertissement :** Pour autoriser la haute disponibilité dans votre système de production, incluez au moins deux unités de couplage dans PREFLIST pour les structures utilisées par IBM MQ.

## Procédure

1. Actualisez la règle CFRM à l'aide de la commande suivante :

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN1
```

Modèle de règle CFRM pour la structure IBM1SCEN1 :

```
STRUCTURE
NAME(IBM1SCEN1)
SIZE(1024M)
INITSIZE(512M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
```

2. Vérifiez que la structure a été créée correctement en utilisant la commande suivante :

```
D XCF,STR,STRNAME=IBM1SCEN1
```

À ce stade, votre structure n'a pas été allouée, affichée par la ligne STATUS, au groupe de partage de files d'attente.

3. Configurez IBM MQ de sorte qu'il utilise la structure définie dans la règle CFRM.

- a. Utilisez la commande `DEFINE CFSTRUCT` avec le nom de structure SCEN1 pour créer un objet CFSTRUCT IBM MQ :

```
DEFINE CFSTRUCT(SCEN1)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 1')
```

```
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFLD1SZ(64K) OFFLD1TH(70)
OFFLD2SZ(64K) OFFLD2TH(80)
OFFLD3SZ(64K) OFFLD3TH(90)
```

- b. Validez la structure à l'aide de la commande `DISPLAY CFSTRUCT`.
- c. Définissez la file d'attente partagée `SCEN1.Q` de sorte qu'elle utilise la structure `SCEN1` à l'aide de la commande `MQSC` :

```
DEFINE QLOCAL(SCEN1.Q) QSGDISP(SHARED) CFSTRUCT(SCEN1) MAXDEPTH(999999999)
```

4. Utilisez IBM MQ Explorer pour insérer un message unique dans la file d'attente `SCEN1.Q` et retirer à nouveau le message.
5. Lancez la commande suivante pour vérifier que la structure est maintenant allouée :

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Dans la sortie de la commande, vérifiez que la ligne `STATUS` affiche `ALLOCATED`.

## Résultats

Vous avez créé la configuration de base. Vous pouvez maintenant vous faire une idée des performances de base de votre configuration en utilisant la méthode de votre choix.

## Que faire ensuite

Ajout du fichier de messages partagé et de la mémoire de classe d'archivage à la structure initiale

### Concepts associés

«Utilisation de la mémoire de classe d'archivage avec des files d'attente partagées», à la page 199

L'utilisation de la mémoire de classe d'archivage, combinée à des files d'attente partagées IBM MQ for z/OS, peut présenter des avantages.

 *Ajout du fichier de messages partagé et de la mémoire de classe d'archivage à la structure initiale*

Comment ajouter le fichier de messages partagé et la mémoire de classe d'archivage pour l'archivage d'urgence dans IBM MQ.

## Pourquoi et quand exécuter cette tâche

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

Cette partie de la tâche utilise la configuration de base décrite dans «Archivage d'urgence - configuration de base», à la page 208. Le scénario décrit l'ajout de fichiers de messages partagés (SMDS) puis de la mémoire de classe d'archivage (SCM) à la structure initiale.

Cette configuration finale est illustrée dans la [Figure 63](#), à la page 211.

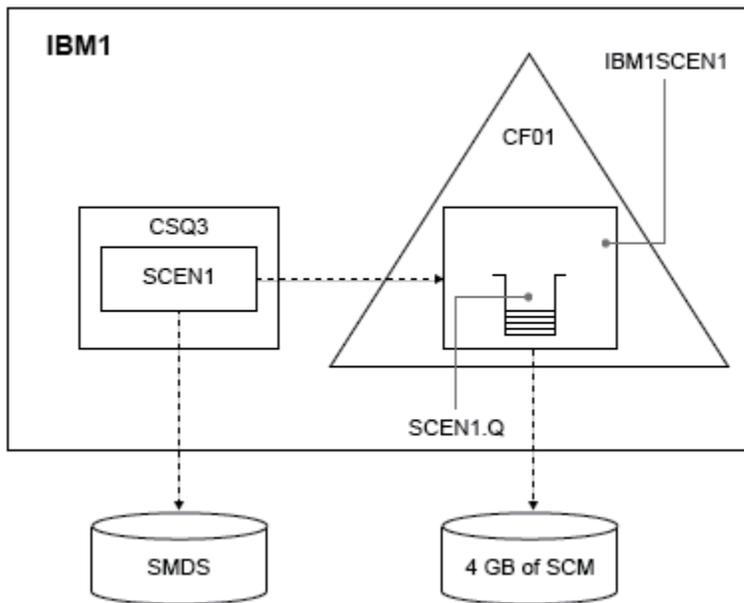


Figure 63. Ajout à la configuration du fichier de messages partagé et de la mémoire de classe d'archivage pour l'archivage d'urgence

## Procédure

1. Créez le fichier de messages partagé utilisé par la structure d'application SCEN1 en éditant le modèle de code JCL **CSQ4SMDS**, comme suit :

```
//CSQ4SMDS JOB NOTIFY=&SYSUID
//*
/* Allocate SMDS
/*
//DEFINE EXEC PGM=IDCAMS,REGION=4M
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER          -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS) -
MEGABYTES(5000 3000) -
LINEAR                  -
SHAREOPTIONS(2 3) ) -
DATA                    -
(NAME(CSQSMDS.SCEN1.CSQ3.SMDS.DATA) )
/*
/*
/* Format the SMDS
/*
//FORM EXEC PGM=CSQJUFMT,COND=(0,NE),REGION=0M
//STEPLIB DD DSN=MQ800.SCSQANLE,DISP=SHR
// DD DSN=MQ800.SCSQAUTH,DISP=SHR
//SYSUT1 DD DISP=OLD,DSN=CSQSMDS.SCEN1.CSQ3.SMDS
//SYSPRINT DD SYSOUT=*
```

2. Exécutez la commande **ALTER CFSTRUCT** pour modifier la structure d'application SCEN1, afin d'utiliser le fichier SMDS pour le déchargement, en implémentant les règles de déchargement par défaut:

```
ALTER CFSTRUCT(SCEN1) OFFLOAD(SMDS) OFFLD1SZ(32K) OFFLD2SZ(4K) OFFLD3SZ(0K)
DSGROUP('CSQSMDS.SCEN1.*.SMDS') DSBLOCK(1M)
```

Notez ce qui suit :

- SCEN1.Q étant la seule file d'attente partagée dans la structure d'application SCEN1, la valeur **DSBLOCK** est définie sur 1M, qui est la valeur la plus élevée possible. Ce réglage est le plus efficace dans le cadre de ce scénario.

- Les messages envoyés par l'application d'insertion ayant une taille de 30 ko, le déchargement dans le fichier de messages partagé ne commence pas tant que la deuxième règle de déchargement n'est pas satisfaite, c'est-à-dire lorsque la structure est remplie à 80 %.
3. Exécutez à nouveau votre application de test.  
Notez le nombre accru de messages stockés dans la file d'attente.
  4. Ajoutez 4 Go de mémoire de classe d'archivage à la structure IBM1SCEN1 en procédant comme suit :
    - a) Déterminez la quantité de mémoire de classe d'archivage qui est installée et allouée à CF01 en lançant la commande suivante :

```
D CF,CFNAME=CF01
```

- b) Consultez les figures STORAGE - CLASS MEMORY dans la section STORAGE CONFIGURATION de la sortie affichée pour voir le stockage disponible.
- c) Mettez à jour la règle CFRM avec les mots clés SCMMAXSIZE et SCMALGORITHM comme indiqué ci-dessous :

```
STRUCTURE
NAME(IBM1SCEN1)
SIZE(1024M)
INITSIZE(512M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
SCMMAXSIZE(4G)
SCMALGORITHM(KEYPRIORITY1)
```

5. Activez la règle CFRM en lançant la commande suivante :

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=polname
```

6. Régénérez la structure IBM1SCEN1.

Cette opération est nécessaire car la structure a été allouée au moment où vous avez apporté les modifications précédentes.

Lancez la commande suivante pour régénérer la structure :

```
SETXCF START,REBUILD,STRNM=IBM1SCEN1
```

## Résultats

Vous avez ajouté avec succès la mémoire de classe d'archivage à votre configuration.

## Que faire ensuite

Optimisez les performances de votre système. Pour plus d'informations, voir [«Optimisation de l'utilisation de la mémoire de classe d'archivage»](#), à la page 212.

 *Optimisation de l'utilisation de la mémoire de classe d'archivage*

Comment améliorer l'utilisation de la mémoire de classe d'archivage (SCM).

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou télécharger des messages dans le fichier SMDS.

Exécutez ensuite la commande suivante :

```
D XCF,STR,STRNAME=IBM1SCEN1
```

Comme la structure était déjà pleine de données de message, à la suite des tests précédents, une partie de la régénération comportait une étape de pré-transfert de certains messages depuis la structure vers la mémoire de classe d'archivage. Ce processus a été lancé à l'aide de la commande précédente.

Exemple de sortie de cette commande :

```
ACTIVE STRUCTURE
-----
ALLOCATION TIME: 06/17/2014 09:28:50
CFNAME : CF01
COUPLING FACILITY: 002827.IBM.02.00000000B8D7
PARTITION: 3B CPCID: 00
STORAGE CONFIGURATION ALLOCATED MAXIMUM %
ACTUAL SIZE: 1024 M 1024 M 100
AUGMENTED SPACE: 3 M 142 M 2
STORAGE-CLASS MEMORY: 88 M 4096 M 2
ENTRIES: 120120 1089536 11
ELEMENTS: 240240 15664556 1
SPACE USAGE IN-USE TOTAL %
ENTRIES: 84921 219439 38
ELEMENTS: 2707678 3149050 85
EMCS: 2 282044 0
LOCKS: 1024
SCMHIGHTHRESHOLD : 90
SCMLOWTHRESHOLD : 70
ACTUAL SUBNOTIFYDELAY: 5000
PHYSICAL VERSION: CD5186A0 2BD8885C
LOGICAL VERSION: CD515C50 CE2ED258
SYSTEM-MANAGED PROCESS LEVEL: 9
XCF GRPNAME : IXCL0053
DISPOSITION : KEEP
ACCESS TIME : NOLIMIT
MAX CONNECTIONS: 32
# CONNECTIONS : 1
CONNECTION NAME ID VERSION SYSNAME JOBNAME ASID STATE
-----
CSQEIBM1CSQ301 01 00010059 SC61 CSQ3MSTR 0091 ACTIVE
```

Notez les éléments suivants dans la sortie de la commande :

- STORAGE\_CLASS MEMORY confirme qu'un **MAXIMUM** de 4096 Mo de mémoire de classe d'archivage a été ajouté à la structure.
- La valeur ALLOCATED affectée à la quantité de STORAGE-CLASS MEMORY utilisée pour le pré-transfert. La structure comporte maintenant de l'espace disponible alors qu'il n'en restait plus avant l'ajout de la mémoire de classe d'archivage.
- La quantité de AUGMENTED SPACE utilisée pour suivre l'utilisation de la mémoire de classe d'archivage.
- Le point au niveau duquel l'algorithme de pré-transfert commence à transférer les données depuis la structure vers la mémoire de classe d'archivage lorsque la structure est pleine à 90 %. Ceci est indiqué par la propriété **SCMHIGHTHRESHOLD** non configurable.
- Le point au-dessous duquel l'algorithme de pré-extraction commence à transférer les données de la mémoire de classe d'archivage vers la structure correspond au moment où cette dernière est pleine à 70 %. Ceci est indiqué par la propriété **SCMLOWTHRESHOLD** non configurable.

Vous pouvez maintenant tester les différentes manières d'optimiser l'utilisation de la mémoire de classe d'archivage. Notez ce qui suit :

- Une fois que la mémoire de classe d'archivage a été utilisée pour stocker les messages, vous ne pouvez plus modifier la structure tant que vous n'avez pas supprimé toutes les données contenues dans la mémoire de classe d'archivage.

Dans ce cas, cela signifie que le rapport entrées/éléments est figé à la valeur définie lors de la première utilisation de la mémoire de classe d'archivage. Vous devez veiller à ce que la structure est à l'état voulu avant que l'algorithme de pré-transfert commence à transférer les données vers la mémoire de classe d'archivage.

- La taille de la structure actuelle est-elle correcte avant d'utiliser la mémoire de classe d'archivage ?

Par exemple, avez-vous augmenté le paramètre **INITSIZE** de 512 Mo à 1 Go ?

Si vous n'effectuez pas cette opération, il est possible, bien que la structure soit configurée pour l'auto-modification, que l'algorithme de pré-transfert commence à transférer les données vers la mémoire de classe d'archivage avant que la modification ne puisse démarrer. La structure est alors figée en utilisant 512 Mo de mémoire réelle.

- Le rapport entrées/éléments est-il correct avant d'utiliser la mémoire de classe d'archivage ?

L'objectif de ce scénario est d'augmenter le nombre de pointeurs de messages déchargés qui peuvent être stockés dans la structure et dans la mémoire de classe d'archivage dans son ensemble, et de conserver entièrement autant de messages que possible dans la mémoire de la structure. L'accès à ces messages est plus rapide que via un fichier de messages partagé.

Vous devez donc disposer d'une structure qui commence avec un rapport entrées/éléments satisfaisant pour le stockage des messages, puis transite par un rapport satisfaisant pour le stockage des pointeurs de message avant que l'algorithme de pré-transfert ne démarre pour la première fois. Cette transition peut être réalisée, en partie, en utilisant les règles de déchargement IBM MQ.

Modifiez les règles de déchargement en lançant la commande suivante :

```
ALTER CFSTRUCT(SCEN1) OFFLD1SZ(0K)
```

Plusieurs exécutions peuvent s'avérer nécessaires pour optimiser les rapports entrées/éléments.

Le tableau suivant présente les améliorations pouvant être apportées au nombre de messages insérés dans la file d'attente au cours des différentes phases du scénario d'archivage d'urgence.

Description du test	Nombre de messages	Délai de remplissage de la file d'attente (secondes)
Configuration de base	27 850	3.2
Fichier de messages partagé avec règles de déchargement par défaut	205 000	158
Mémoire de classe d'archivage avec règles de déchargement par défaut	828 610	469
Mémoire de classe d'archivage avec règles de déchargement ajustées	1 135 775	679

La dernière colonne du tableau indique que l'ajustement des règles de déchargement a eu l'effet requis.

Vous devez examiner votre système pour déterminer si vous pouvez améliorer ces chiffres d'une manière ou d'une autre. Par exemple, il se peut que vous manquiez d'espace disponible dans le fichier de messages partagé. Si vous pouvez allouer plus d'espace à ce dernier, vous devriez pouvoir augmenter de manière significative le nombre de messages dans la file d'attente.

 *Gain de performances - configuration de base*

Comment configurer un scénario de base pour un gain de performances dans IBM MQ.

## Pourquoi et quand exécuter cette tâche

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou télécharger des messages dans le fichier SMDS.

Ce scénario décrit l'utilisation de la mémoire de classe d'archivage pour augmenter le nombre de messages pouvant être stockés dans une file d'attente partagée sans subir le coût en terme de performance que représente l'utilisation du fichier de messages partagé.

Ce scénario initial est très similaire à celui utilisé pour l'archivage d'urgence et il utilise les éléments ci-dessous.

- Un groupe de partage de files d'attente, IBM1, qui contient un gestionnaire de files d'attente unique, CSQ3. Outre la structure d'administration, le groupe de partage de files d'attente a défini une structure d'application unique, SCEN2.
- L'unité de couplage (CF) CF01, dans laquelle la structure d'application SCEN2 est stockée en tant que structure IBM1SCEN2. Cette structure a la taille maximale de 2 Go.
- Une file d'attente partagée unique, SCEN2.Q, qui est configurée pour utiliser la structure d'application.

Cette configuration est illustrée dans la [Figure 64](#), à la page 215.

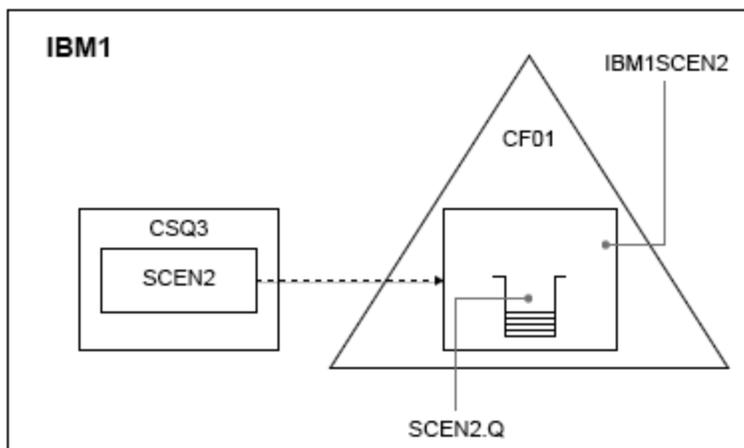


Figure 64. Configuration de base

Supposons également que le gestionnaire de files d'attente CSQ3 est déjà le seul membre du groupe de partage de files d'attente IBM1.

Vous devez ajouter la définition pour la structure IBM1SCEN2 à la règle CFRM (Coupling Facility Resource Manager). Dans un souci de simplicité, la structure est définie de manière à pouvoir être créée dans une seule unité de couplage, CF01, en spécifiant PREFLIST (CF01).

Modèle de règle CFRM pour la structure IBM1SCEN2 :

```
STRUCTURE
NAME(IBM1SCEN2)
SIZE(2048M)
INITSIZE(2048M)
ALLOWAUTOALT(YES)
FULLTHRESHOLD(85)
PREFLIST(CF01)
ALLOWREALLOCATE(YES)
DUPLEX(DISABLED)
ENFORCEORDER(NO)
```

Les mots clés **INITSIZE** et **SIZE** ont tous les deux la valeur 2048M afin que la structure ne puisse pas être redimensionnée.

## Procédure

1. Actualisez la règle CFRM à l'aide de la commande suivante :

```
SETXCF START,POLICY,TYPE=CFRM,POLNAME=IBM1SCEN2
```

2. Vérifiez que la structure a été créée correctement en utilisant la commande suivante :

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Le lancement de la commande précédente génère la sortie suivante :

```
RESPONSE=SC61
IXC360I 07.58.51 DISPLAY XCF 581
STRNAME: IBM1SCEN2
STATUS: NOT ALLOCATED
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY

EVENT MANAGEMENT: MESSAGE-BASED   MANAGER SYSTEM NAME: SC53
MANAGEMENT LEVEL : 01050107
```

À ce stade, votre structure n'a pas été allouée, affichée par la ligne STATUS, au groupe de partage de files d'attente.

3. Configurez IBM MQ de sorte qu'il utilise la structure définie dans la règle CFRM.
  - a. Utilisez la commande `DEFINE CFSTRUCT` avec le nom de structure SCEN2 pour créer un objet CFSTRUCT IBM MQ.

```
DEFINE CFSTRUCT(SCEN2)
CFCONLOS(TOLERATE)
CFLEVEL(5)
DESCR('Structure for SCM scenario 2')
RECOVER(NO)
RECAUTO(YES)
OFFLOAD(DB2)
OFFFLD1SZ(64K) OFFFLD1TH(70)
OFFFLD2SZ(64K) OFFFLD2TH(80)
OFFFLD3SZ(64K) OFFFLD3TH(90)
```

- b. Vérifiez la structure à l'aide de la commande `DISPLAY CFSTRUCT`.
- c. Définissez la file d'attente partagée SCEN2.Q de sorte qu'elle utilise la structure SCEN2 à l'aide de la commande MQSC :

```
DEFINE QLOCAL(SCEN2.Q) QSGDISP(SHARED) CFSTRUCT(SCEN2) MAXDEPTH(999999999)
```

4. Utilisez IBM MQ Explorer pour insérer un message unique dans la file d'attente SCEN2.Q et retirer à nouveau le message.
5. Lancez la commande suivante pour vérifier que la structure est maintenant allouée :

```
D XCF,STR,STRNAME=IBM1SCEN2
```

Vérifiez la sortie de la commande, dont une partie est affichée, et assurez-vous que la ligne STATUS comporte la valeur ALLOCATED.

```
RESPONSE=SC61
IXC360I 08.31.27 DISPLAY XCF 703
STRNAME: IBM1SCEN2
```

```
STATUS: ALLOCATED
EVENT MANAGEMENT: MESSAGE-BASED
TYPE: SERIALIZED LIST
POLICY INFORMATION:
POLICY SIZE : 2048 M
POLICY INITSIZE: 2048 M
POLICY MINSIZE : 1536 M
FULLTHRESHOLD : 85
ALLOWAUTOALT : YES
REBUILD PERCENT: N/A
DUPLEX : DISABLED
ALLOWREALLOCATE: YES
PREFERENCE LIST: CF01
ENFORCEORDER : NO
EXCLUSION LIST IS EMPTY
```

De plus, notez les valeurs des zones figurant dans la section SPACE USAGE :

- ENTRIES
- ELEMENTS
- EMCS
- LOCKS

Exemple de valeurs :

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	344686	345242	99
ELEMENTS:	6548455	6548467	99
EMCS:	2	780318	0
LOCKS:	1024		

## Résultats

Vous avez créé la configuration de base. Vous pouvez maintenant vous faire une idée des performances de base de votre configuration en utilisant la méthode de votre choix.

## Que faire ensuite

Vous devez tester le scénario de base. En tant qu'exemple, vous pouvez utiliser les trois applications suivantes, en les lançant dans l'ordre indiqué et en les exécutant simultanément.

1. Utilisez une application PCF pour demander la valeur de taille actuelle ( **CURDEPTH** ) de SCEN2 . Q toutes les cinq secondes. La sortie peut être utilisée pour tracer la taille de la file d'attente au fil du temps.
2. Une application d'extraction dotée d'une seule unité d'exécution extrait de manière répétée des messages de SCEN2 . Q en utilisant une commande `get` avec un temps d'attente infini. Pour simuler le traitement des messages qui ont été supprimés, l'application d'extraction fait une pause de quatre millisecondes tous les dix messages supprimés.
3. Une application d'insertion dotée d'une seule unité d'exécution insère au total un million de messages de 4 ko non persistants dans SCEN2 . Q. Cette application n'effectue pas de pause entre chaque insertion de message. Les messages sont donc insérés dans SCEN2 . Q plus vite que l'application d'extraction ne peut les extraire.

Ainsi, lorsque l'application d'insertion est exécutée, la taille de SCEN2 . Q augmente.

Lorsque la structure IBM1SCEN2 est remplie et que l'application d'insertion reçoit le code raison `MQRC_STORAGE_MEDIUM_FULL`, cette dernière se met en veille pendant cinq secondes avant de tenter d'insérer le message suivant dans la file d'attente.

Vous pouvez tracer les résultats de l'application `CURDEPTH` sur une certaine période de temps. Vous obtenez une sortie sous forme de courbe en dents de scie lorsque l'application d'insertion fait une pause pour laisser la file d'attente se vider partiellement.

Accédez à [«Ajout de la mémoire de classe d'archivage à la structure initiale»](#), à la page 218.

## Concepts associés

«Utilisation de la mémoire de classe d'archivage avec des files d'attente partagées», à la page 199  
L'utilisation de la mémoire de classe d'archivage, combinée à des files d'attente partagées IBM MQ for z/OS, peut présenter des avantages.

**z/OS** Ajout de la mémoire de classe d'archivage à la structure initiale

Comment ajouter la mémoire de classe d'archivage pour un gain de performance dans IBM MQ.

## Pourquoi et quand exécuter cette tâche

**Important :** IBM z16 doit être la dernière génération de IBM Z<sup>®</sup> pour prendre en charge l'utilisation de la mémoire flash virtuelle (également appelée mémoire de classe de stockage ou SCM) pour les images d'unité de couplage. Pour plus d'informations, voir: [IBM Z et IBM LinuxONE 4Q 2023 Statements of Direction](#).

Vous pouvez également utiliser des structures plus grandes ou décharger des messages dans le fichier SMDS.

Cette partie de la tâche utilise la configuration de base décrite dans «Gain de performances - configuration de base», à la page 214. Le scénario décrit l'ajout de la mémoire de classe d'archivage (SCM) à la structure initiale.

Cette configuration finale est illustrée dans la [Figure 65](#), à la page 218.

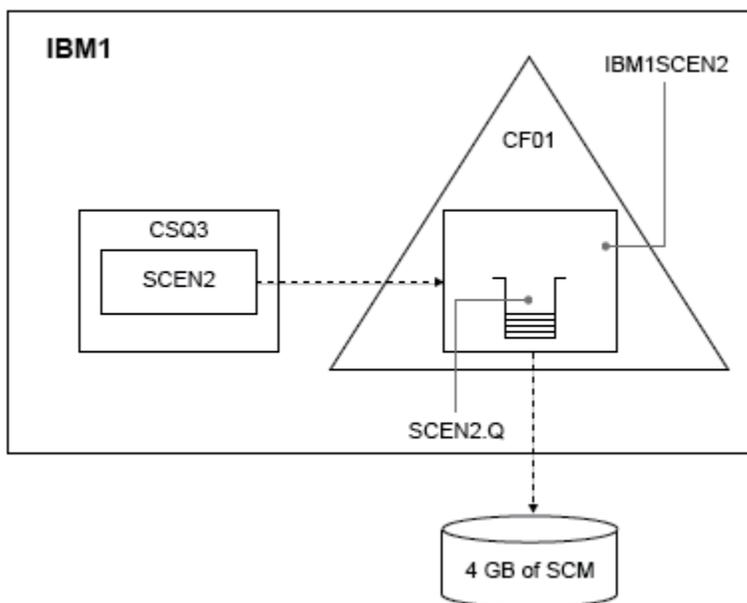


Figure 65. Configuration de l'ajout de la mémoire de classe d'archivage pour un gain de performance

## Procédure

1. Ajoutez 4 Go de mémoire de classe d'archivage à la structure IBM1SCEN2 en procédant comme suit :
  - a) Déterminez la quantité de mémoire de classe d'archivage qui est installée et allouée à CF01 en lançant la commande suivante :

```
D CF,CFNAME=CF01
```

- b) Consultez les figures STORAGE-CLASS MEMORY dans la section STORAGE CONFIGURATION de la sortie affichée pour voir le stockage disponible.
- c) Mettez à jour la règle CFRM avec les mots clés SCMMAXSIZE et SCMALGORITHM comme indiqué ci-dessous :

```
STRUCTURE
NAME (IBM1SCEN2)
SIZE (2048M)
INITSIZE (2048M)
ALLOWAUTOALT (YES)
FULLTHRESHOLD (85)
PREFLIST (CF01)
ALLOWREALLOCATE (YES)
DUPLEX (DISABLED)
ENFORCEORDER (NO)
SCMMAXSIZE (4G)
SCMALGORITHM (KEYPRIORITY1)
```

2. Activez la règle CFRM en lançant la commande suivante :

```
SETXCF START ,POLICY ,TYPE=CFRM ,POLNAME=IBM1SCEN2
```

3. Régénérez la structure IBM1SCEN2.

Cette opération est nécessaire car la structure a été allouée au moment où vous avez apporté les modifications précédentes.

Lancez la commande suivante pour régénérer la structure :

```
SETXCF START ,REBUILD ,STRNM=IBM1SCEN2
```

4. Lancez la commande suivante pour confirmer la nouvelle configuration de la structure :

```
D XCF ,STR ,STRNAME=IBM1SCEN2
```

Vérifiez la sortie de la commande, dont une partie est affichée ci-dessous :

SPACE USAGE	IN-USE	TOTAL	%
ENTRIES:	33	342684	0
ELEMENTS:	48	6503697	0
EMCS:	2	575600	0
LOCKS:		1024	

## Résultats

Calculez la modification de l'utilisation de la mémoire réelle par l'augmentation de la mémoire de contrôle nécessaire à l'utilisation de la mémoire de classe d'archivage.

- Avant l'ajout de la mémoire de classe d'archivage, la structure a les totaux suivants, comme indiqué dans [«Gain de performances - configuration de base»](#), à la page 214 :
  - 345 242 entrées
  - 6 548 467 éléments
  - 780 318 EMCS
- Après l'ajout de la mémoire de classe d'archivage, la structure a les totaux suivants :
  - 342 684 entrées
  - 6 503 697 éléments
  - 575 600 EMCS

En utilisant ces chiffres, après l'ajout de la mémoire de classe d'archivage, la taille de la structure est réduite de :

- 2558 entrées
- 44 770 éléments
- 204 718 EMCS

La quantité de mémoire de la structure utilisée pour gérer la mémoire de classe d'archivage est la suivante pour une structure de 2 Go avec 4 Go de mémoire de classe d'archivage allouée :

$$(2558 + 44,770 + 204,718) * 256 = 61.5 \text{ MB}$$

Notez que l'ajout de mémoire de classe d'archivage supplémentaire ne peut que réduire de manière marginale la taille de la structure car la quantité de mémoire de contrôle utilisée pour suivre la mémoire de classe d'archivage augmente, à mesure que la taille de la structure et la quantité de mémoire de classe d'archivage allouée augmentent.

## Que faire ensuite

Répétez les tests décrits dans la section finale de [«Gain de performances - configuration de base»](#), à la page 214.

Vous pouvez tracer les résultats de l'application révisée sur une certaine période de temps. En comparant le tracé à celui que vous avez obtenu précédemment, vous obtenez une sortie avec sans courbe en dents de scie car l'application d'insertion n'attend plus que la file d'attente se vide partiellement.

Pour plus d'informations, voir [MP16: WebSphere MQ for z/OS -Planification et optimisation de la capacité](#).

## Mise en file d'attente répartie et groupes de partage de files d'attente

La mise en file d'attente répartie et les groupes de partage de files d'attente sont les deux techniques que vous pouvez utiliser pour améliorer la disponibilité des systèmes d'application. Cette rubrique permet de rechercher des informations supplémentaires concernant ces techniques.

Destiné à compléter la haute disponibilité des messages sur les files d'attente partagées, le composant de mise en file d'attente répartie d'IBM MQ est doté des fonctions supplémentaires suivantes :

- Disponibilité plus élevée pour le réseau.
- Augmentation de la capacité des connexions réseau entrantes vers le groupe de partage de files d'attente.

La [Figure 66](#), à la [page 221](#) illustre la mise en file d'attente répartie et les groupes de partage de files d'attente. Elle présente deux gestionnaires de files d'attente dans un sysplex. Ces deux gestionnaires appartiennent au même groupe de partage de files d'attente. Ils peuvent tous deux accéder à la file d'attente partagée SQ1. Les gestionnaires de files d'attente du réseau (sur AIX et Windows, par exemple) peuvent placer les messages dans cette file d'attente via l'initiateur de canal de l'un des deux gestionnaires. Les applications clonées des deux gestionnaires de files d'attente traitent la file d'attente.

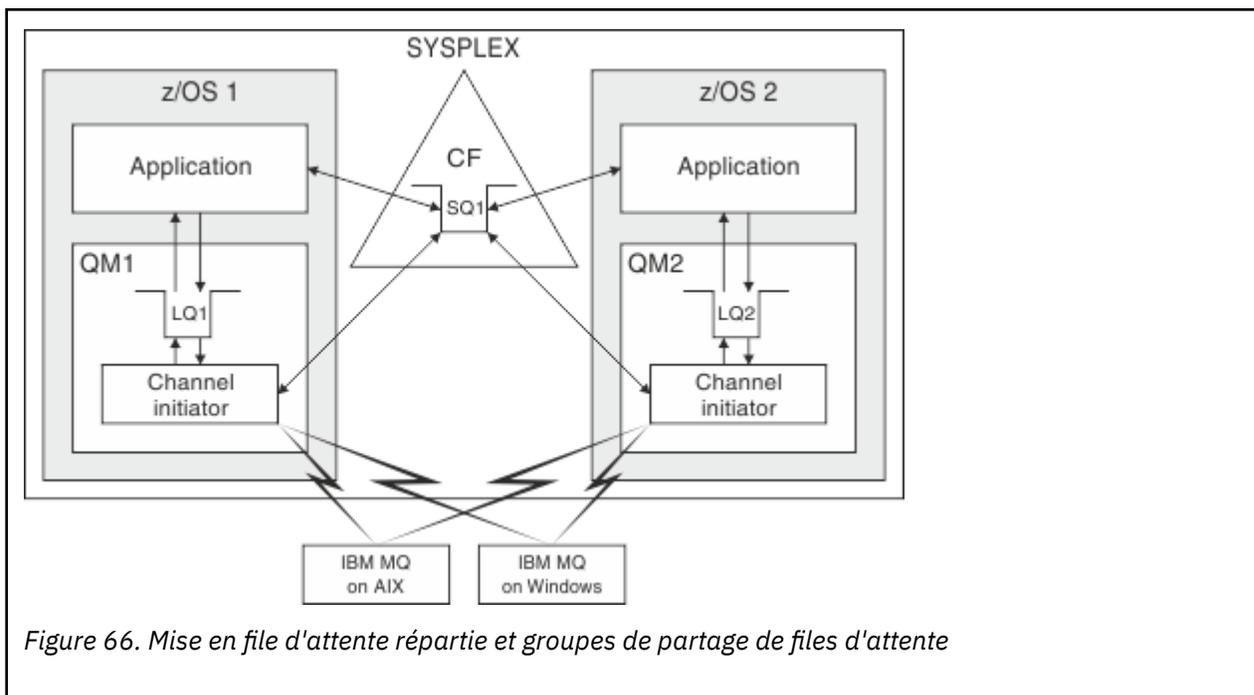


Figure 66. Mise en file d'attente répartie et groupes de partage de files d'attente

### Concepts associés

«Canaux partagés», à la page 221

Cette rubrique explique les concepts relatifs aux canaux partagés et leur utilisation avec IBM MQ for z/OS.

«Mise en file d'attente intragroupe», à la page 223

Les files d'attente intra-groupe permettent de transférer des messages entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente.

«Clusters et groupes de partage de files d'attente», à la page 224

Consultez cette rubrique pour comprendre comment utiliser les groupes de partage de files d'attente avec les clusters.

### z/OS Canaux partagés

Cette rubrique explique les concepts relatifs aux canaux partagés et leur utilisation avec IBM MQ for z/OS.

Certains produits de réseau proposent un mécanisme permettant de masquer du réseau les incidents liés au serveur ou d'équilibrer les demandes de réseau entrantes entre plusieurs serveurs. Ces produits mettent un *port générique* à la disposition des demandes de connexion réseau entrantes et les demande entrantes peuvent être satisfaites via une connexion à l'un de ces serveurs.

Ces produits de réseau sont les suivants :

- ressources génériques VTAM
- SYSPLEX Distributor

L'initiateur de canal tire parti de ces produits pour utiliser les fonctions des files d'attente partagées

Il existe deux types de canaux partagés : les *canaux de communications entrantes partagés* et les *canaux de communications sortantes partagés*.

- [Canaux de communications entrantes partagés](#)
- [Canaux de communications sortantes partagés](#)

Pour plus d'informations sur les canaux, consultez

- [Récapitulatif sur les canaux partagés](#)
- [Statut du canal partagé](#)

## Canaux de communications entrantes partagés

Chaque initiateur de canal du groupe de partage de files d'attente démarre une tâche d'écoute supplémentaire pour être à l'écoute sur un *port générique*. Ce port est mis à la disposition du serveur par l'une des technologies de prise en charge (VTAM, TCP/IP). La technologie utilisée dans le réseau répartit les demandes entrantes d'association de réseau au port générique vers les programmes d'écoute du groupe de partage de files d'attente qui sont à l'écoute sur ce port.

Vous pouvez démarrer un canal sur l'initiateur de canal vers lequel l'association entrante est dirigée si cet initiateur a accès à une définition de canal pour un canal de ce nom. Vous pouvez définir une définition de canal comme étant privée pour un gestionnaire de files d'attente ou comme étant stockée dans le référentiel partagé et donc disponible partout (définition globale). Vous pouvez donc faire en sorte qu'une définition de canal soit disponible sur n'importe quel initiateur de canal du groupe de partage en la définissant en tant que définition globale.

Autre différence en cas de démarrage d'un canal via le port générique : la synchronisation se fait avec le groupe de partage de files d'attente et non avec un gestionnaire de files d'attente donné. Prenez l'exemple d'un gestionnaire de files d'attente éloignées qui démarre un canal via le port générique. Lorsque le canal démarre pour la première fois, il peut démarrer sur le gestionnaire de files d'attente QM1 et les messages sont transmis. Si le canal s'arrête, puis redémarre sur le gestionnaire de files d'attente QM2, les informations sur le nombre de messages transmis sont toujours correctes car la synchronisation se fait avec le groupe de partage de files d'attente.

Vous pouvez utiliser un canal de communications entrantes démarré via le port générique pour placer les messages dans une file d'attente. Le gestionnaire de files d'attente éloignées ne sait pas si la file d'attente cible est partagée ou non. Si oui, le gestionnaire de files d'attente éloignées se connecte via un initiateur de canal de manière à respecter l'équilibre des charges et les messages sont placés dans la file d'attente partagée.

Si la file d'attente cible est privée, les messages sont placés dans la file d'attente privée appartenant au gestionnaire de files d'attente auquel l'instance actuelle du canal est connectée. Dans cet environnement, qualifié de *files d'attente locales répliquées*, les mêmes files d'attente privées doivent être définies pour chaque gestionnaire de files d'attente.

## Configuration des canaux SVRCONN pour un groupe de partage de files d'attente

La configuration optimale des canaux SVRCONN dans un groupe de partage de files d'attente consiste à définir des programmes d'écoute privés sur chaque initiateur de canal utilisant un numéro de port différent à partir des canaux point-à-point. Ces ports d'écoute sont alors utilisés en tant que ressources d'arrière-plan pour un nouveau mécanisme de distribution de la charge de travail tel que Sysplex Distributor à l'aide d'adresses IP virtuelles. Ces adresses sont alors utilisées en tant qu'adresses cible pour les définitions CLNTCONN du réseau. Le canal SVRCONN peut être défini avec QSGDISP(GROUP) de sorte que la même définition est disponible pour tous les gestionnaires de files d'attente du groupe de partage. Cette configuration évite l'utilisation d'un programme d'écoute partagé et réduit par conséquent l'effet sur les performances du groupe de partage de files d'attente qui gère l'état des canaux partagés, ce qui n'est pas nécessaire pour les canaux client/serveur.

## Canaux de communications sortantes partagés

Un canal de communications sortantes est considéré comme un canal partagé s'il reçoit des messages d'une file d'attente de transmission partagée. Un canal partagé conserve les informations de synchronisation au niveau du groupe de partage de files d'attente. Cela signifie qu'il est possible de redémarrer ce canal sur un autre gestionnaire de files d'attente et sur une autre instance d'initiateur de canal du groupe de partage de files d'attente en cas d'incident lié au sous-système de communications, à l'initiateur de canal ou au gestionnaire de files d'attente. Ce type de redémarrage est une fonction des canaux partagés appelée *reprise par canal homologue*.

## Équilibrage de charge pour les canaux de communications sortantes partagés

Un canal partagé sortant peut démarrer sur n'importe quel initiateur de canal du groupe de partage de files d'attente si vous n'avez pas spécifié que vous souhaitez qu'il démarre sur un initiateur particulier. L'initiateur de canal sélectionné par IBM MQ est déterminé selon les critères suivants :

- Le sous-système de communications requis est-il actuellement disponible pour l'initiateur de canal ?
- Existe-t-il une connexion Db2 disponible pour l'initiateur de canal ?
- Quel est l'initiateur de canal dont la charge de travail est la plus faible ? La charge de travail inclut les canaux actifs et les canaux qui sont en train de procéder à une nouvelle tentative.

## Récapitulatif sur les canaux partagés

Les différences entre les canaux partagés et les canaux privés sont énumérés ci-dessous :

### Canal privé

Il est lié à un initiateur de canal unique.

- Le canal de communications sortantes utilise une file d'attente de transmission locale.
- Le canal de communications entrantes démarre via un port local.
- Les informations de synchronisation sont contenues dans la file d'attente SYSTEM.CHANNEL.SYNCQ.

### Canal partagé

La charge de travail est équilibrée avec la haute disponibilité.

- Le canal de communications sortantes utilise une file d'attente de transmission partagée.
- Le canal de communications entrantes démarre via un port générique.
- Les informations de synchronisation sont contenues dans la file d'attente SYSTEM.QSG.CHANNEL.SYNCQ.

Vous indiquez si un canal est privé ou partagé lorsque vous démarrez les canaux à l'aide des options CHLDISP associées à la commande `START CHANNEL`. Vous pouvez démarrer un canal partagé en le déclenchant de la même manière qu'un canal privé. Toutefois, lors du démarrage d'un canal partagé, IBM MQ procède à l'équilibrage de la charge de travail et démarre le canal sur l'initiateur de canal le plus approprié dans le groupe de partage de files d'attente. (Vous pouvez si nécessaire spécifier qu'un canal partagé doit démarrer sur un initiateur de canal particulier.)

## Statut du canal partagé

Les initiateurs de canal d'un groupe de partage de files d'attente gèrent une table de statut des canaux dans Db2. Cette table enregistre quels canaux sont actifs sur quels initiateurs de canal. La table de statut des canaux partagés est utilisée en cas d'incident lié à un initiateur de canal ou au système de communications. Elle indique quels canaux doivent être redémarrés sur un initiateur de canal différent du groupe de partage des files d'attente.

### **Mise en file d'attente intragroupe**

Les files d'attente intra-groupe permettent de transférer des messages entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente.

Vous pouvez procéder à des transferts rapides de messages entre différents gestionnaires de files d'attente d'un groupe de partage sans définir de canal. Cette procédure utilise la file d'attente système appelée SYSTEM.QSG.TRANSMIT.QUEUE, qui est une file d'attente de transmission partagée. Chaque gestionnaire de files d'attente du groupe de partage démarre une tâche appelée agent de mise en file d'attente intra-groupe, qui attend que les messages destinés à leur gestionnaire de files d'attente arrivent

dans cette file d'attente. Lorsqu'un tel message est détecté, il est supprimé de la file d'attente et placé dans la file d'attente de destination correcte.

Les règles de résolution de nom standard sont utilisées, mais si la fonction de mise en file d'attente intra-groupe est activée et si le gestionnaire de files d'attente cible se trouve dans le groupe de partage de files d'attente, SYSTEM.QSG.TRANSMIT.QUEUE est utilisée pour transférer le message au gestionnaire de files d'attente de destination correct et aucune file d'attente de transmission ni aucun canal n'est utilisé.

Vous pouvez activer la fonction de mise en file d'attente intra-groupe via un attribut du gestionnaire de files d'attente. Cette fonction déplace les messages non persistants à l'extérieur du point de synchronisation et les messages persistants dans le point de synchronisation. Si un problème survient lors de la distribution des messages à la file d'attente cible, la fonction de mise en file d'attente intra-groupe tente de les placer dans la file d'attente de rebut. Si celle-ci est saturée ou non définie, les messages non persistants sont supprimés, mais les messages persistants sont annulés et renvoyés à SYSTEM.QSG.TRANSMIT.QUEUE et l'agent IGQ tente à nouveau de les distribuer jusqu'à ce qu'il y parvienne.

Un canal partagé entrant recevant un message destiné à une file d'attente se trouvant sur un autre gestionnaire de files d'attente du groupe de partage peut utiliser la fonction de mise en file d'attente intra-groupe pour *transmettre* le message à la destination voulue.

Dans certains cas, vous souhaitez peut-être que le gestionnaire de files d'attente local place un message directement dans la file d'attente cible, s'il s'agit d'une file d'attente partagée, au lieu de le transférer d'abord au gestionnaire de files d'attente cible. Vous pouvez utiliser l'attribut du gestionnaire de files d'attente SQQMNAME pour contrôler cette opération. Si vous définissez la valeur de SQQMNAME sur USE, la commande MQOPEN est exécutée sur le gestionnaire de files d'attente défini par ObjectQMgrName. Toutefois, si la file d'attente cible est partagée, si vous définissez la valeur IGNORE pour SQQMNAME et si le nom ObjectQMgrName est celui d'un autre gestionnaire de files d'attente du groupe de partage, la file d'attente partagée est ouverte sur le gestionnaire de files d'attente local. Si le gestionnaire de files d'attente local ne parvient pas à ouvrir la file d'attente cible ou à placer un message dans la file d'attente, ce message est transféré au nom ObjectQMgrName défini via un canal IGQ ou MQ.

La fonction de mise en file d'attente intra-groupe prend en charge les messages de grande taille, la taille maximale étant 100 Mo *moins* la longueur de l'en-tête de la file d'attente de transmission.

Si vous utilisez cette fonction, les utilisateurs doivent bénéficier du même accès aux files d'attente sur chaque gestionnaire de files d'attente du groupe de partage.

## **Clusters et groupes de partage de files d'attente**

Consultez cette rubrique pour comprendre comment utiliser les groupes de partage de files d'attente avec les clusters.

Vous pouvez mettre vos files d'attente partagées à la disposition d'un cluster dans une définition unique. Pour cela, vous devez définir le nom du cluster lorsque vous définissez la file d'attente partagée.

Pour les utilisateurs du réseau, la file d'attente partagée apparaît comme étant hébergée par chaque gestionnaire de files d'attente du groupe de partage de files d'attente (la file d'attente partagée n'est pas annoncée comme étant hébergée par le groupe de partage de files d'attente). Les clients peuvent démarrer des sessions avec un membre du groupe de files d'attente pour placer les messages dans la même file d'attente partagée.

La [Figure 67](#), à la [page 225](#) présente l'accès des membres d'un cluster à une file d'attente partagée via n'importe quel membre du groupe de partage de files d'attente.

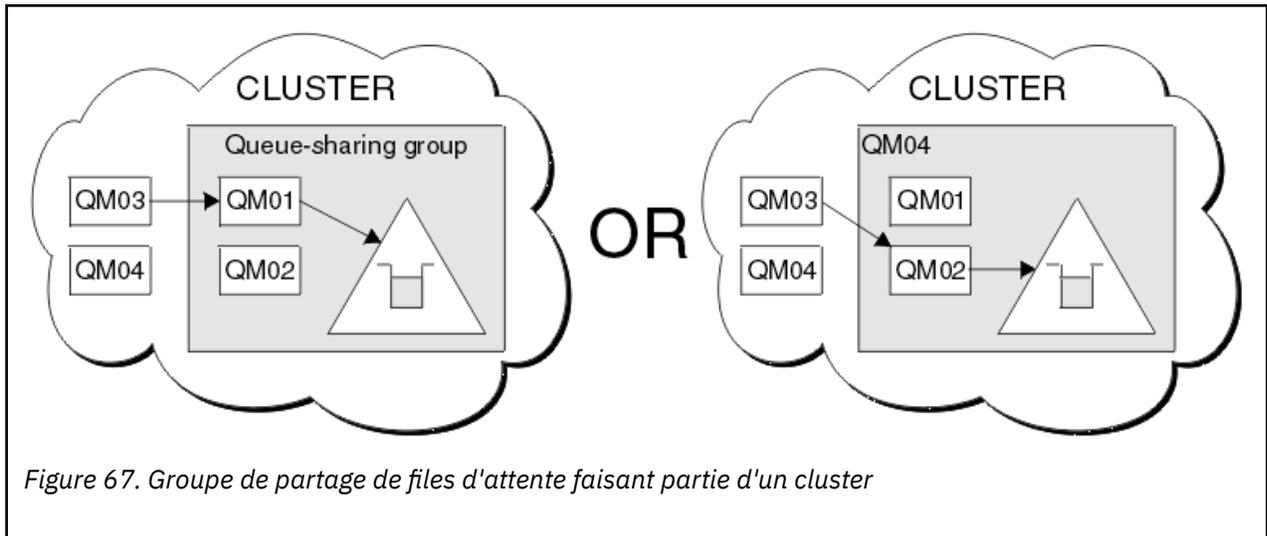


Figure 67. Groupe de partage de files d'attente faisant partie d'un cluster

## > z/OS Influence de la distribution de charge de travail avec des files d'attente partagées

Cette rubrique permet de comprendre comment certains facteurs liés aux files d'attente partagées ont une incidence sur la distribution de la charge de travail dans un groupe de partage de files d'attente.

IBM MQ ne fournit pas de fonction d'équilibrage de charge pour les files d'attente partagées. Toutefois, la distribution de la charge de travail dans un groupe de partage de files d'attente peut être *basée sur l'extraction*. Le choix du gestionnaire de files d'attente qui va traiter une file d'attente (c'est-à-dire qui va recevoir un message écrit dans une file d'attente partagée) est déterminé par la capacité de traitement disponible de chaque gestionnaire de files d'attente du groupe de partage et par les objectifs en matière de gestion de la charge de travail définis dans le sysplex.

Il est cependant important de prendre en compte le fait que le gestionnaire de files d'attente qui exécute l'appel MQPUT d'un message peut également largement contribuer à décider quel gestionnaire de files d'attente va extraire le message.

### Probabilité plus grande que le gestionnaire de files d'attente local exécute l'appel MQGET

Pour une application exécutant un appel MQPUT, le gestionnaire de files d'attente local auquel l'application est connectée est considéré comme le gestionnaire local.

L'identification du gestionnaire de files d'attente qui traite l'appel MQPUT d'un message en exécutant un appel MQGET au nom d'une application ayant lancé la commande d'extraction est déterminée par les considérations suivantes.

Lorsqu'un message est placé dans une file d'attente partagée vide, le gestionnaire de files d'attente local est généralement posté avant qu'un autre gestionnaire de files d'attente du groupe de partage de files d'attente ne soit notifié. Si le gestionnaire de files d'attente local est en position de traiter le message, il reçoit une notification signalant une transition provenant de l'unité de couplage avant les autres gestionnaires de files d'attente du groupe de partage. (Une notification signalant une transition est une notification indiquant que la file d'attente partagée est passée de l'état vide à non vide.)

Dans ce cas, les scénarios possibles sont les suivants :

1. Appel MQPUT du point non synchronisé du message non persistant et *insertion rapide dans une méthode d'accès get en attente*.

S'il existe une application avec un appel *MQGET avec attente* sur le gestionnaire de files d'attente local pour la file d'attente, l'appel MQPUT du message est transmis directement au tampon de l'application ayant émis la commande d'extraction et n'est pas écrite dans la file d'attente. Cela est vrai pour les files d'attente partagées et non partagées. Cette fonction est souvent appelée mécanisme d'*insertion*

*rapide dans une méthode d'accès get en attente.* Dans le cas d'une file d'attente partagée, aucun autre gestionnaire de files d'attente du groupe de partage n'est notifié, car aucune file d'attente ne passe de l'état vide à non vide. Par exemple, cela signifie que si ce gestionnaire de files d'attente peut traiter toutes les commandes d'insertion à partir de cette application et en supposant qu'aucune autre application n'insère de message dans la file d'attente, aucun autre gestionnaire de files d'attente dans le groupe de partage de files d'attente ne participe au nettoyage de la file d'attente. Cependant, s'il n'existe aucun appel MQGET avec attente dans le gestionnaire de files d'attente local et qu'un message est inséré dans la file d'attente partagée, l'unité de couplage notifie les autres gestionnaires de files d'attente du groupe de partage de files d'attente conformément à ses règles en matière de notification et de signalement de transition.

2. Appel MQPUT d'un message persistant ou d'un message dans un point de synchronisation.

Dans ce cas, s'il existe une application avec un appel MQGET avec attente dans le gestionnaire de files d'attente local, le message est inséré dans la file d'attente partagée et l'unité de couplage notifie les autres gestionnaires de files d'attente du groupe de partage de files d'attente selon ses règles en matière de notification et de signalement de transition. Toutefois, le gestionnaire de files d'attente local n'attend pas la notification de transition de l'unité de couplage, mais commence par traiter un appel MQGET avec attente local et exécute généralement l'extraction de ce message au nom de l'application avant qu'un autre gestionnaire de files d'attente du groupe de partage de files d'attente ne puisse répondre à une notification de l'unité de couplage. Cette opération dépend du niveau d'occupation du gestionnaire de files d'attente local. Sinon, le gestionnaire de files d'attente notifié par l'unité de couplage en raison de l'arrivée du message dans la file d'attente vide va essayer de traiter d'abord la commande d'extraction. Le premier gestionnaire de files d'attente qui répond traite le nouveau message.

3. Enfin, si la file d'attente n'est pas nettoyée de ses messages et si l'unité de couplage a envoyé une notification signalant que l'état de celle-ci est passé de vide à non vide, tous les gestionnaires de files d'attente connectés peuvent participer à son traitement. Dans ce cas, la charge de travail est considérée comme étant basée sur l'extraction.

Cette conception permet d'améliorer les performances de la distribution par rapport à une distribution de la charge de travail purement basée sur l'extraction. L'objectif est de tirer parti de la haute disponibilité offerte par les files d'attente se trouvant dans l'unité de couplage tout en permettant au gestionnaire de files d'attente, lorsque c'est possible, d'exécuter l'appel MQGET sans devoir faire référence à l'unité de couplage et donc de traiter les messages de façon aussi efficace que possible.

D'autres approches peuvent être adoptées lorsque l'équilibrage de la charge est considéré comme plus important que les améliorations des performances décrites précédemment. C'est le cas lorsque vous souhaitez vérifier qu'aucune application ayant émis une commande d'extraction n'est connectée au gestionnaire de files d'attente auquel l'application ayant émis la commande d'insertion est connectée. Avec cette conception, tous les messages sont placés dans la file d'attente et tous les gestionnaires de files d'attente du groupe de partage sont notifiés lorsque la file d'attente passe de l'état vide à non vide, conformément à l'algorithme utilisé par l'unité de couplage pour traiter ces transitions. En outre, le mécanisme d'insertion rapide dans une méthode d'accès get en attente n'est pas applicable.

**z/OS** **Où trouver des informations supplémentaires sur les files d'attente partagées et les groupes de partage de files d'attente ?**

Le tableau ci-dessous fournit des informations supplémentaires sur la façon dont IBM MQ for z/OS utilise les files d'attente partagées et les groupes de partage de files d'attente.

Tableau 18. Où trouver des informations supplémentaires sur les files d'attente partagées et les groupes de partage de files d'attente ?	
Topic	Section
Récupération de groupe de partage de files d'attente	«Reprise et redémarrage sur z/OS», à la page 270

Tableau 18. Où trouver des informations supplémentaires sur les files d'attente partagées et les groupes de partage de files d'attente ? (suite)

Topic	Section
Sécurité de groupe de partage de files d'attente	<a href="#">«Concepts de sécurité dans IBM MQ for z/OS», à la page 289</a>
Définition d'objets globaux et privés Orientation des commandes vers différents gestionnaires de files d'attente	<a href="#">Emission de commandes</a>
Planification de votre unité de couplage de couplage	<a href="#">Définition des ressources d'unité de couplage</a>
Planification de l'environnement SMDS	<a href="#">Planification de votre environnement de fichiers de messages partagés</a>
Planification de votre Environnement Db2	<a href="#">Planification de votre environnement Db2</a>
Configuration des paramètres système de files d'attente partagées	<a href="#">«Files d'attente partagées et groupes de partage de files d'attente», à la page 176</a>
programmes utilitaires des programmes utilitaires	<a href="#">Utilisation des utilitaires IBM MQ</a>
Messages de console	<a href="#">Messages pour IBM MQ for z/OS</a>
Commandes MQSC	<a href="#">Les commandes MQSC</a>
Clusters IBM MQ	<a href="#">Configuration d'un cluster de gestionnaires de files d'attente</a>
Mise en file d'attente répartie IBM MQ Noms de canal	<a href="#">Présentation de la gestion répartie des files d'attente</a>
Applications d'écriture	<a href="#">Présentation de la conception des applications</a>
Appel MQCONN	<a href="#">MQCONN</a>

z/OS

## Mise en file d'attente intragroupe

Cette section décrit la mise en files d'attente intra-groupe, une fonction IBM MQ for z/OS propre à la plateforme z/OS. Cette fonction n'est disponible que pour les gestionnaires de files d'attente définis dans un groupe de partage de files d'attente.

Pour plus d'informations sur les groupes de partage de files d'attente, voir [«Files d'attente partagées et groupes de partage de files d'attente», à la page 176](#).

## z/OS concepts de la mise en file d'attente intra-groupe

La mise en file d'attente intra-groupe permet de distribuer plus efficacement de petits messages à des files d'attente résidant sur des gestionnaires de files d'attente éloignées dans un groupe de partage de files d'attente.

La mise en file d'attente intra-groupe permet le transfert potentiellement rapide et moins onéreux de petits messages entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente, sans qu'il soit nécessaire de définir des canaux entre les gestionnaires de files d'attente.

Le diagramme ci-dessous présente un exemple type de mise en file d'attente intra-groupe.

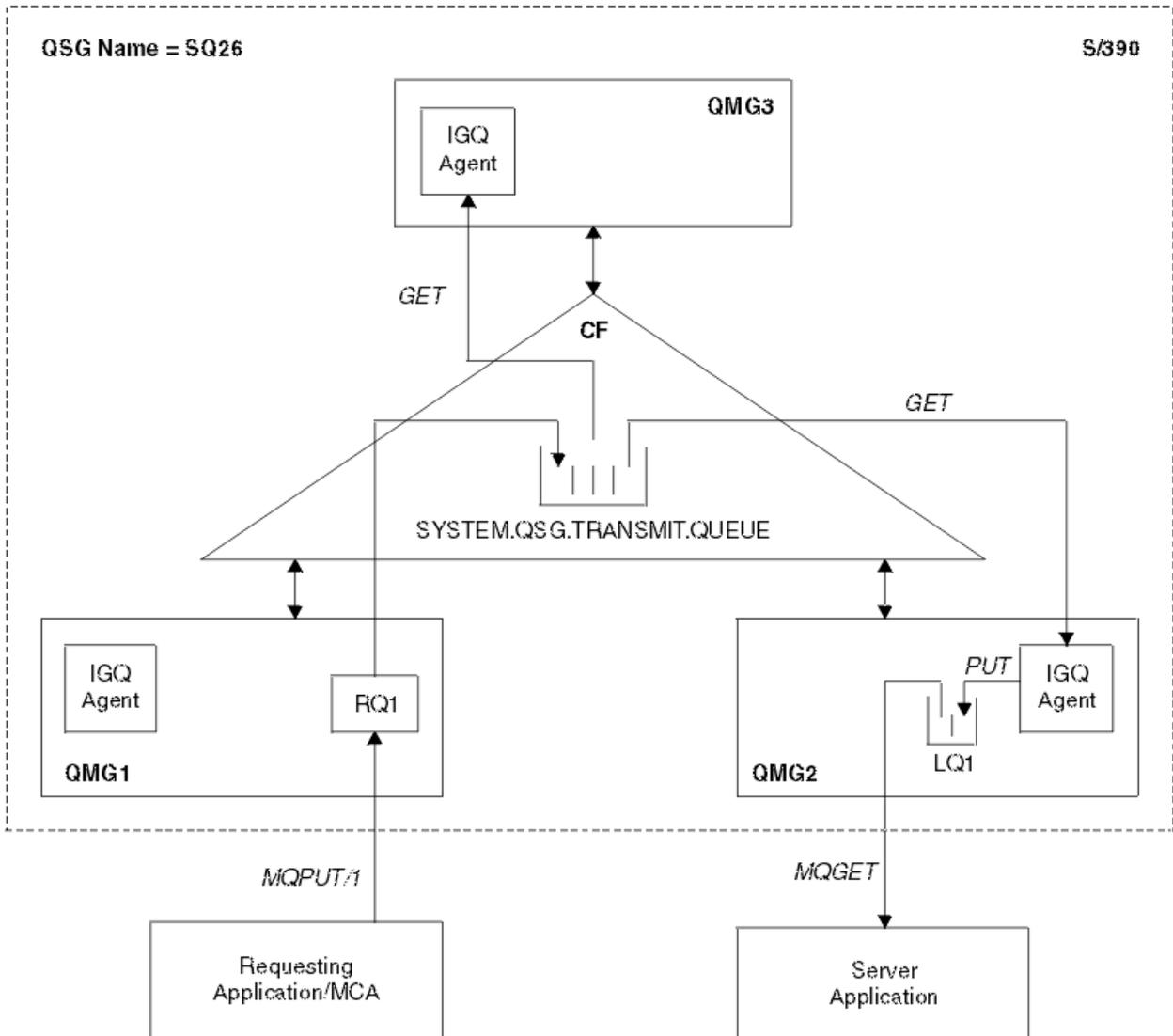


Figure 68. Exemple de mise en file d'attente intra-groupe

Le diagramme présente :

- Des agents de mise en file d'attente intra-groupe exécutés sur trois gestionnaires de files d'attente (QMG1, QMG2 et QMG3), qui sont définis dans un groupe de partage de files d'attente appelé SQ26.
- La file d'attente de transmission SYSTEM.QSG.TRANSMIT.QUEUE définie dans l'unité de couplage.
- Une définition de file d'attente éloignée définie dans le gestionnaire de files d'attente QMG1.
- Une file d'attente locale définie dans le gestionnaire de files d'attente QMG2.
- Une application de demande (il peut s'agir d'un agent MCA (Message Channel Agent)) connectée au gestionnaire de files d'attente QMG1.

- Une application serveur connectée au gestionnaire de files d'attente QMG2.
- Un message de demande placé dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

## Mise en file d'attente intra-groupe et agent de mise en file d'attente intra-groupe

Un agent de mise en file d'attente intra-groupe est démarré lors de l'initialisation d'un gestionnaire de files d'attente. Lorsque des applications ouvrent et insèrent des messages dans des files d'attente éloignées, le gestionnaire de files d'attente local détermine si la mise en file d'attente intra-groupe est utilisée pour le transfert des messages. Si la mise en file d'attente intra-groupe doit être utilisée, le gestionnaire de files d'attente local place le message dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE. L'agent de mise en file d'attente intra-groupe sur le gestionnaire de files d'attente éloignées cible extrait le message et le place dans la file d'attente de destination.

### Terminologie de la mise en file d'attente intra-groupe

Explications de la terminologie : mise en file d'attente intra-groupe, file d'attente de transmission partagée pour utilisation par la mise en file d'attente intra-groupe et agent de mise en file d'attente intra-groupe.

### Mise en file d'attente intragroupe

La mise en file d'attente intra-groupe permet le transfert potentiellement rapide et moins onéreux de messages entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente, sans qu'il ne soit nécessaire de définir des canaux.

### File d'attente de transmission partagée pour utilisation par la mise en file d'attente intra-groupe

Chaque groupe de partage de files d'attente comporte une file d'attente de transmission partagée appelée SYSTEM.QSG.TRANSMIT.QUEUE qui peut être utilisée par la mise en file d'attente intra-groupe. Si cette dernière est activée, la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE apparaît dans le même chemin de résolution de nom lors de l'ouverture des files d'attente éloignées. Lorsque des applications, y compris les agents MCA (Message Channel Agent), insèrent des messages dans une file d'attente éloignée, le gestionnaire de files d'attente local détermine si les messages sont éligibles pour un transfert rapide, puis les place dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

### Agent de mise en file d'attente intra-groupe

L'agent de mise en file d'attente intra-groupe correspond à la tâche, démarrée lors de l'initialisation du gestionnaire de files d'attente, qui attend l'arrivée de messages appropriés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE. Cet agent extrait les messages appropriés de cette file d'attente puis les distribue aux files d'attente de destination.

L'agent de mise en file d'attente intra-groupe est toujours démarré car la mise en file d'attente intra-groupe est utilisée par le gestionnaire de files d'attente proprement dit pour son propre traitement interne.

### Avantages de la mise en file d'attente intra-groupe

Voici les avantages de la mise en file d'attente intra-groupe : définitions de système réduites, administration de système réduite, performances améliorées, prise en charge de la migration et distribution de messages lors de l'utilisation de plusieurs tronçons entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente.

Les avantages de la mise en file d'attente intra-groupe sont les suivants :

#### Définitions de système réduites

Avec la mise en file d'attente intra-groupe, il n'est plus nécessaire de définir des canaux entre les gestionnaires de files d'attente dans un groupe de partage de files d'attente.

## Administration de système réduite

Dans la mesure où aucun canal n'est défini entre les gestionnaires de files d'attente dans un groupe de partage de files d'attente, l'administration de canal n'est pas requise.

## Performances améliorées

Etant donné qu'un seul agent mise en file d'attente intra-groupe est requis pour la distribution d'un message à une file d'attente cible (à la place de deux agents émetteur et récepteur intermédiaires), la distribution des messages à l'aide de la mise en file d'attente intra-groupe peut être moins onéreuse que la distribution des messages à l'aide de canaux. La mise en file d'attente intra-groupe ne comporte qu'un composant récepteur, car le composant émetteur n'est plus nécessaire. Une telle économie est due au fait que le message est mis à la disposition de l'agent de mise en file d'attente intra-groupe sur le gestionnaire de files d'attente de destination à des fins de distribution à la file d'attente de destination une fois que l'opération d'extraction sur le gestionnaire de files d'attente local est terminée et, dans le cas des messages insérés dans le cadre de point de synchronisation, validée.

## Prise en charge de la migration

Les applications externes à un groupe de partage de files d'attente peuvent distribuer des messages à une file d'attente résidant sur n'importe quel gestionnaire de files d'attente du groupe de partage de files d'attente, tout en étant connectées uniquement à un gestionnaire de files d'attente particulier du groupe de partage de files d'attente. En effet, les messages arrivant sur un canal récepteur, qui sont destinés à une file d'attente sur un gestionnaire de files d'attente éloignées, peuvent être envoyés de façon transparente à la file d'attente de destination à l'aide de la mise en file d'attente intra-groupe. Cette fonction permet le déploiement des applications dans le groupe de partage de files d'attente sans qu'il ne soit nécessaire de modifier les systèmes externes au groupe de partage de files d'attente.

Une configuration type est illustrée dans le diagramme suivant où :

- Une application de demande connectée au gestionnaire de files d'attente QMG1 doit envoyer un message à une file d'attente du gestionnaire de files d'attente QMG3.
- Le gestionnaire de files d'attente QMG1 n'est connecté qu'au gestionnaire de files d'attente QMG2.
- Les gestionnaires de files d'attente QMG2 et QMG3, qui ont précédemment été connectés à l'aide de canaux, font désormais partie du groupe de partage de files d'attente SQ26.

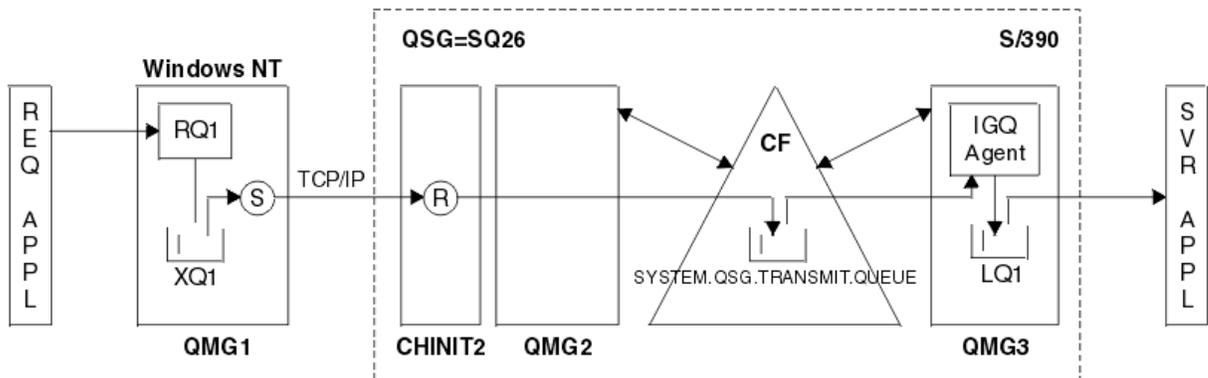


Figure 69. Exemple de prise en charge de la migration

Le flux des opérations est le suivant :

1. L'application de demande insère un message, destiné à la file d'attente locale LQ1 sur le gestionnaire de files d'attente éloignées QMG3, dans la définition de file d'attente éloignée RQ1.
2. Le gestionnaire de files d'attente QMG1, exécuté sur un poste de travail Windows NT place le message dans la file d'attente de transmission XQ1.
3. L'agent MCA émetteur (S) sur QM1 transmet le message, à l'aide de TCP/IP, à l'agent MCA récepteur (R) sur l'initiateur de canal CHINIT2.
4. L'agent MCA récepteur (R) sur l'initiateur de canal CHINIT2 place le message dans la file d'attente de transmission partagée SYSTEM.QSG.TRANSMIT.QUEUE.

5. L'agent de mise en file d'attente intra-groupe sur le gestionnaire de files d'attente QMG3 extrait le message de la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE, puis le place dans la file d'attente locale cible LQ1.
6. L'application serveur extrait le message de la file d'attente locale cible puis le traite.

### **Distribution de messages lors de l'utilisation de plusieurs tronçons entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente.**

Le diagramme précédent de la section [Prise en charge de la migration](#) illustre également la distribution de messages lors de l'utilisation de plusieurs tronçons entre les gestionnaires de files d'attente d'un groupe de partage de files d'attente. Les messages arrivant sur un gestionnaire de files d'attente dans le groupe de partage de files d'attente mais destinés à une file d'attente sur un autre gestionnaire de files d'attente dans le groupe de partage de files d'attente peuvent être facilement transmis à la file d'attente de destination sur le gestionnaire de files d'attente de destination, à l'aide de la mise en file d'attente intra-groupe.

## **z/OS Limitations de la mise en file d'attente intra-groupe**

Les limitations de la mise en file d'attente intra-groupe sont les suivantes : messages éligibles pour un transfert à l'aide de la mise en file d'attente intra-groupe, nombre d'agents de mise en file d'attente intra-groupe par gestionnaire de files d'attente et démarrage et arrêt de l'agent de mise en file d'attente intra-groupe.

Cette rubrique décrit les limitations de la mise en file d'attente intra-groupe.

### **Messages éligibles pour un transfert à l'aide de la mise en file d'attente intra-groupe**

Dans la mesure où la mise en file d'attente intra-groupe utilise une file d'attente de transmission partagée définie dans l'unité de couplage, la mise en file d'attente intra-groupe est limitée à la distribution des messages dont la longueur correspond à la longueur maximale prise en charge pour les files d'attente moins la longueur d'un en-tête de file d'attente de transmission (MQXQH).

### **Nombre d'agents de mise en file d'attente intra-groupe par gestionnaire de files d'attente**

Un seul agent de mise en file d'attente intra-groupe est démarré par gestionnaire de files d'attente dans un groupe de partage de files d'attente.

### **Démarrage et arrêt de l'agent de mise en file d'attente intra-groupe**

L'agent de mise en file d'attente intra-groupe est démarré lors de l'initialisation d'un gestionnaire de files d'attente et arrêté lors de l'arrêt du gestionnaire de files d'attente. Il est conçu pour être une tâche de longue durée à restauration automatique (en cas d'arrêt anormal). Si une erreur se produit au niveau de la définition de la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE (par exemple, si cette file d'attente n'est pas accessible en extraction), l'agent de mise en file d'attente intra-groupe continue de refaire des tentatives. Si l'agent de mise en file d'attente intra-groupe détecte une erreur donnant lieu à l'arrêt normal de l'agent alors que le gestionnaire de files d'attente est toujours actif, il peut être démarré à l'aide d'une commande ALTER QMGR IGQ(ENABLED). Cette commande évite d'avoir à recycler le gestionnaire de files d'attente.

### **Définition de l'attribut de gestionnaire de files d'attente IGQ sur ENABLED ou DISABLED**

Si l'attribut IGQ du gestionnaire de files d'attente est défini sur ENABLED ou DISABLED, les descripteurs d'objet existants peuvent être invalidés avec le code anomalie MQRC\_OBJECT\_CHANGED. Pour plus d'informations, voir [Initiation à la mise en file d'attente intra-groupe](#).

## **z/OS Initiation à la mise en file d'attente intra-groupe**

Vous pouvez activer, désactiver et utiliser la mise en file d'attente intra-groupe comme décrit dans cette rubrique.

### **Activation de la mise en file d'attente intra-groupe**

Pour activer la mise en file d'attente intra-groupe sur vos gestionnaires de files d'attente, procédez comme suit :

- Définissez une file d'attente de transmission partagée appelée SYSTEM.QSG.TRANSMIT.QUEUE. La définition de cette file d'attente se trouve dans thlqual.SCSQPROC(CSQ4INSS), l'exemple de

CSQINP2 pour les objets SYSTEM des groupes de partage de files d'attente. Cette file d'attente doit être définie à l'aide des attributs appropriés, comme indiqué dans thlqual.SCSQPROC(CSQ4INSS), pour que la mise en file d'attente intra-groupe fonctionne correctement.

- Dans la mesure où l'agent de mise en file d'attente intra-groupe est systématiquement démarré lors de l'initialisation du gestionnaire de files d'attente, la mise en file d'attente intra-groupe est toujours disponible pour le traitement de message entrant. L'agent de mise en file d'attente intra-groupe traite tous les messages placés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE. Cependant, pour activer la mise en file d'attente intra-groupe pour le traitement sortant, l'attribut de gestionnaire de files d'attente IGQ doit être paramétré sur ENABLED.

**Important :** Si l'attribut IGQ du gestionnaire de files d'attente est défini sur ENABLED, les descripteurs d'objet existants peuvent être invalidés avec le code anomalie MQR\_OBJECT\_CHANGED. Pour plus d'informations, voir [«Propriétés spécifiques à la mise en file d'attente intra-groupe»](#), à la page 240. Comme indiqué dans la section "Programmer response" pour ce code anomalie, les applications doivent être codées pour gérer cette situation (voir [2041 \(07F9\) \(RC2041\): MQR\\_OBJECT\\_CHANGED](#) pour plus de détails).

De plus, comme IGQ est conçue comme une tâche à exécution de longue durée, capable de reprise automatique, qui démarre lors de l'initialisation et se termine avec l'arrêt, consultez [«Limitations de la mise en file d'attente intra-groupe»](#), à la page 231 pour plus d'informations.

### Désactivation de la mise en file d'attente intra-groupe

Pour désactiver la mise en file d'attente intra-groupe pour le transfert de message sortant, paramétrez l'attribut de gestionnaire de files d'attente IGQ sur DISABLED. Si la mise en file d'attente intra-groupe est désactivée pour un gestionnaire de files d'attente particulier, l'agent de mise en file d'attente intra-groupe sur ce gestionnaire de files d'attente peut néanmoins traiter les messages entrants qui ont été placés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE par un gestionnaire de files d'attente dont la fonction de mise en file d'attente intra-groupe est activée pour le transfert sortant.

**Important :** Si l'attribut IGQ du gestionnaire de files d'attente est défini sur ENABLED, les descripteurs d'objet existants peuvent être invalidés avec le code anomalie MQR\_OBJECT\_CHANGED. Pour plus d'informations, voir [«Propriétés spécifiques à la mise en file d'attente intra-groupe»](#), à la page 240. Comme indiqué dans la section "Programmer response" pour ce code anomalie, les applications doivent être codées pour gérer cette situation (voir [2041 \(07F9\) \(RC2041\): MQR\\_OBJECT\\_CHANGED](#) pour plus de détails).

De plus, comme IGQ est conçue comme une tâche à exécution de longue durée, capable de reprise automatique, qui démarre lors de l'initialisation et se termine avec l'arrêt, consultez [«Limitations de la mise en file d'attente intra-groupe»](#), à la page 231 pour plus d'informations.

### Utilisation de la mise en file d'attente intra-groupe

Une fois la mise en attente intra-groupe activée, elle est disponible à des fins d'utilisation et un gestionnaire de files d'attente l'utilise chaque fois que cela est possible. En d'autres termes, lorsqu'une application insère un message dans une définition de file d'attente éloignée, dans une file d'attente éloignée entièrement qualifiée ou dans une file d'attente de cluster, le gestionnaire de files d'attente détermine si le message est éligible pour être distribué à l'aide de la mise en file d'attente intra-groupe ; si c'est le cas, il place le message dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE. Il n'est pas nécessaire de modifier les applications utilisateur ou les files d'attente d'application car, pour les messages éligibles, le gestionnaire de files d'attente utilise la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE plutôt que toute autre file d'attente de transmission.

## Configurations de la mise en file d'attente intra-groupe

Outre la configuration classique de la mise en file d'attente intra-groupe, d'autres configurations sont possibles.

La [Figure 68](#), à la page 228 décrit la configuration classique.

## Concepts associés

«Mise en files d'attente réparties avec mise en files d'attente intra-groupe (chemins de livraison multiples)», à la page 233

Pour les applications qui traitent des messages de petite taille, il est possible de configurer la mise en file d'attente intra-groupe uniquement pour la distribution de messages entre des gestionnaires de files d'attente d'un groupe de partage de files d'attente.

«Mise en cluster avec mise en file d'attente intra-groupe (plusieurs chemins de distribution)», à la page 235

Il est possible de configurer les gestionnaires de files d'attente de sorte qu'ils résident dans un cluster et dans un groupe de partage de files d'attente.

«Mise en cluster, mise en file d'attente intra-groupe et mise en file d'attente répartie», à la page 238

Il est possible de configurer un gestionnaire de files d'attente qui fait partie d'un cluster et d'un groupe de partage de files d'attente et qui est connecté à un gestionnaire de files d'attente réparties à l'aide d'une paire de canaux émetteur-récepteur.

### **Mise en files d'attente réparties avec mise en files d'attente intra-groupe (chemins de livraison multiples)**

Pour les applications qui traitent des messages de petite taille, il est possible de configurer la mise en file d'attente intra-groupe uniquement pour la distribution de messages entre des gestionnaires de files d'attente d'un groupe de partage de files d'attente.

Le choix de la mise en file d'attente intra-groupes sur les communications de canal peut être contrôlé par le niveau de type CFSTRUCT. (3 au lieu de 4 ou 5). La longueur de message maximale telle que définie sur SYSTEM.QSQ.TRANSMIT.QUEUE.

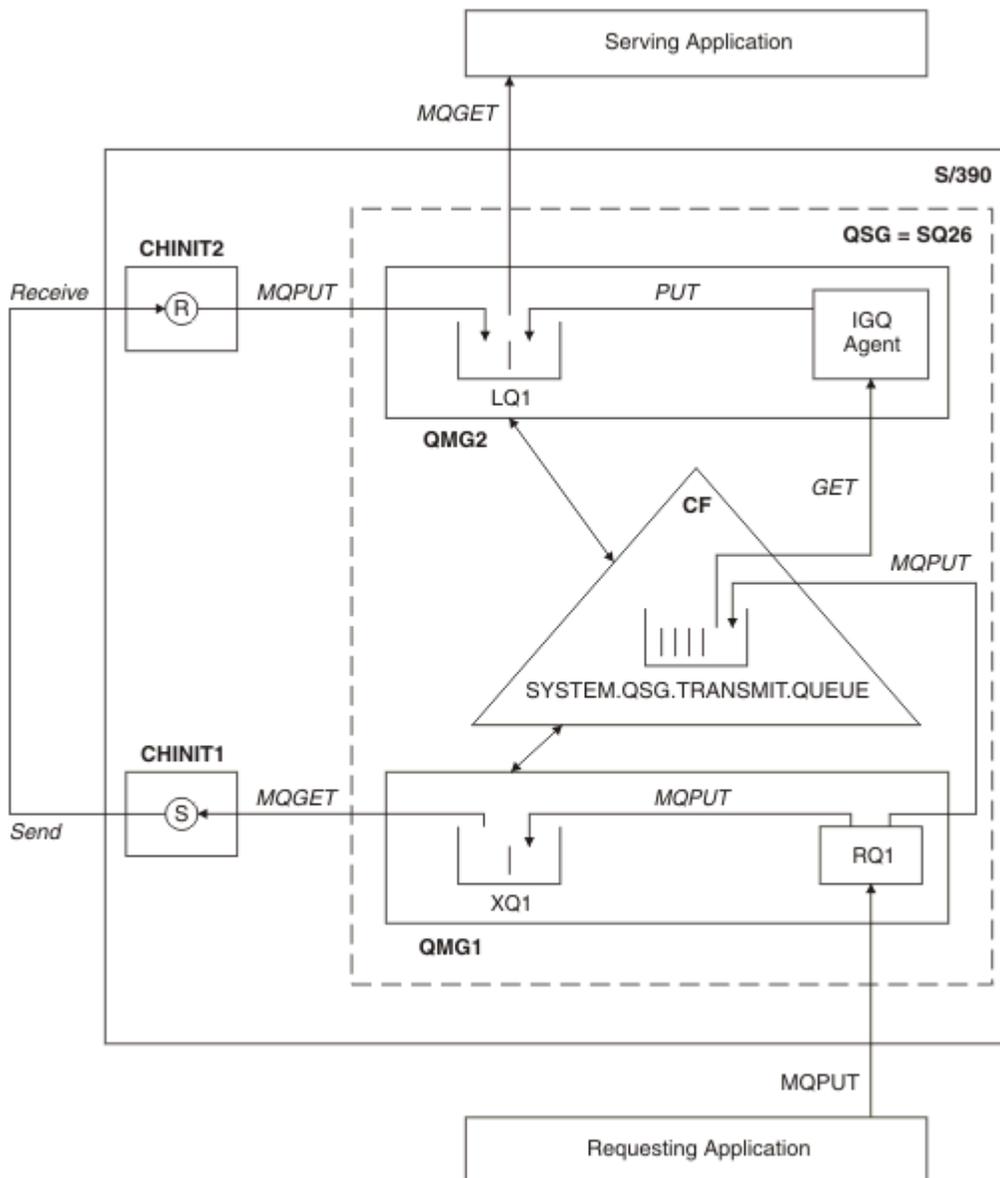


Figure 70. Exemple de configuration

### Processus d'ouverture/insertion

1. Il est à noter que, lorsque l'application de demande ouvre la file d'attente éloignée RQ1, la résolution de nom a lieu pour la file d'attente de transmission non partagée XQ1 et la file d'attente de transmission partagée SYSTEM.QSG.TRANSMIT.QUEUE.
2. Lorsque l'application de demande insère un message sur la file d'attente éloignée, selon si la mise en file d'attente intra-groupe est activée ou non pour le transfert sortant sur le gestionnaire de files d'attente et en fonction des caractéristiques du message, ce dernier est inséré dans la file d'attente de transmission XQ1 ou dans la file d'attente de transmission SYSTEM.QSG.TRANSMIT.QUEUE. Le gestionnaire de files d'attente place tous les messages de grande taille dans la file d'attente de transmission XQ1 et tous les messages de petite taille dans la file d'attente de transmission SYSTEM.QSG.TRANSMIT.QUEUE.
3. Si la file d'attente de transmission XQ1 est saturée ou qu'elle n'est pas disponible, les demandes d'insertion de messages de grande taille échouent de façon synchrone avec un code retour et anomalie approprié. Cependant, les demandes d'insertion de messages de petite taille continuent d'aboutir et sont placées dans la file d'attente de transmission SYSTEM.QSG.TRANSMIT.QUEUE.

4. Si la file d'attente de transmission SYSTEM.QSG.TRANSMIT.QUEUE est saturée ou qu'elle n'est pas accessible en insertion, les demandes d'insertion de messages de petite taille échouent de façon synchrone avec un code retour et anomalie approprié. Cependant, les demandes d'insertion de messages de grande taille continuent d'aboutir et sont placées dans la file d'attente de transmission XQ1. Dans ce cas, aucune tentative d'insertion des messages de petite taille dans une file d'attente de transmission n'est effectuée.

### **Flux des messages de grande taille**

1. L'application de demande insère des messages de grande taille dans la file d'attente éloignée RQ1.
2. Le gestionnaire de files d'attente QMG1 insère les messages dans la file d'attente de transmission XQ1.
3. L'agent MCA émetteur (S) sur le gestionnaire de files d'attente QMG1 extrait les messages de la file d'attente de transmission XQ1 et les envoie au gestionnaire de files d'attente QMG2.
4. L'agent MCA récepteur (R) sur le gestionnaire de files d'attente QMG2 reçoit les messages et les place dans la file d'attente de destination LQ1.
5. L'application de service extrait les messages de la file d'attente LQ1 puis les traite.

### **Flux des messages de petite taille**

1. L'application de demande insère des messages de petite taille dans la file d'attente éloignée RQ1.
2. Le gestionnaire de files d'attente QMG1 insère les messages dans la file d'attente de transmission SYSTEM.QSG.TRANSMIT.QUEUE.
3. La mise en file d'attente intra-groupe sur le gestionnaire de files d'attente QMG2 extrait les messages puis les place dans la file d'attente de destination LQ1.
4. L'application de service extrait les messages de la file d'attente LQ1.

### **Remarques**

1. L'application de demande n'a pas besoin de connaître le mécanisme sous-jacent utilisé pour la distribution des messages.
2. Un mécanisme de distribution de message potentiellement plus rapide peut être obtenu pour les messages de petite taille.
3. Plusieurs chemins sont disponibles pour la distribution de messages (c-à-d, le chemin de canal normal et le chemin de mise en file d'attente intra-groupe).
4. Potentiellement plus rapide, le chemin de mise en file d'attente intra-groupe est préféré au chemin de canal normal. En fonction des caractéristiques de message, la distribution de message peut être répartie sur les deux chemins. Les messages peuvent donc être distribués hors séquence (bien que cette distribution soit également possible si les messages sont distribués à l'aide du chemin de canal normal uniquement).
5. Lorsqu'un chemin a été sélectionné et que des messages ont été placés dans les files d'attente de transmission, seul le chemin sélectionné est utilisé pour la distribution de messages. Les messages non traités dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE ne sont pas détournés vers la file d'attente de transmission XQ1.

### **Mise en cluster avec mise en file d'attente intra-groupe (plusieurs chemins de distribution)**

Il est possible de configurer les gestionnaires de files d'attente de sorte qu'ils résident dans un cluster et dans un groupe de partage de files d'attente.

Lorsque des messages sont envoyés à une file d'attente de cluster et que les gestionnaires de files d'attente de destination locales et éloignées se trouvent dans le même groupe de partage de files d'attente, la mise en file d'attente intra-groupe est utilisée pour la distribution de messages de petite taille (à l'aide de la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE), et la distribution de messages de

grande taille si la mise en files d'attente intra-groupe prend en charge la taille du message. De même, la file d'attente SYSTEM.CLUSTER.TRANSMIT.QUEUE est utilisée pour la distribution de messages à n'importe quel gestionnaire de files d'attente figurant dans le cluster mais hors du groupe de partage de files d'attente. Cette configuration est illustrée dans le diagramme ci-dessous (les initiateurs de canal ne sont pas présentés).

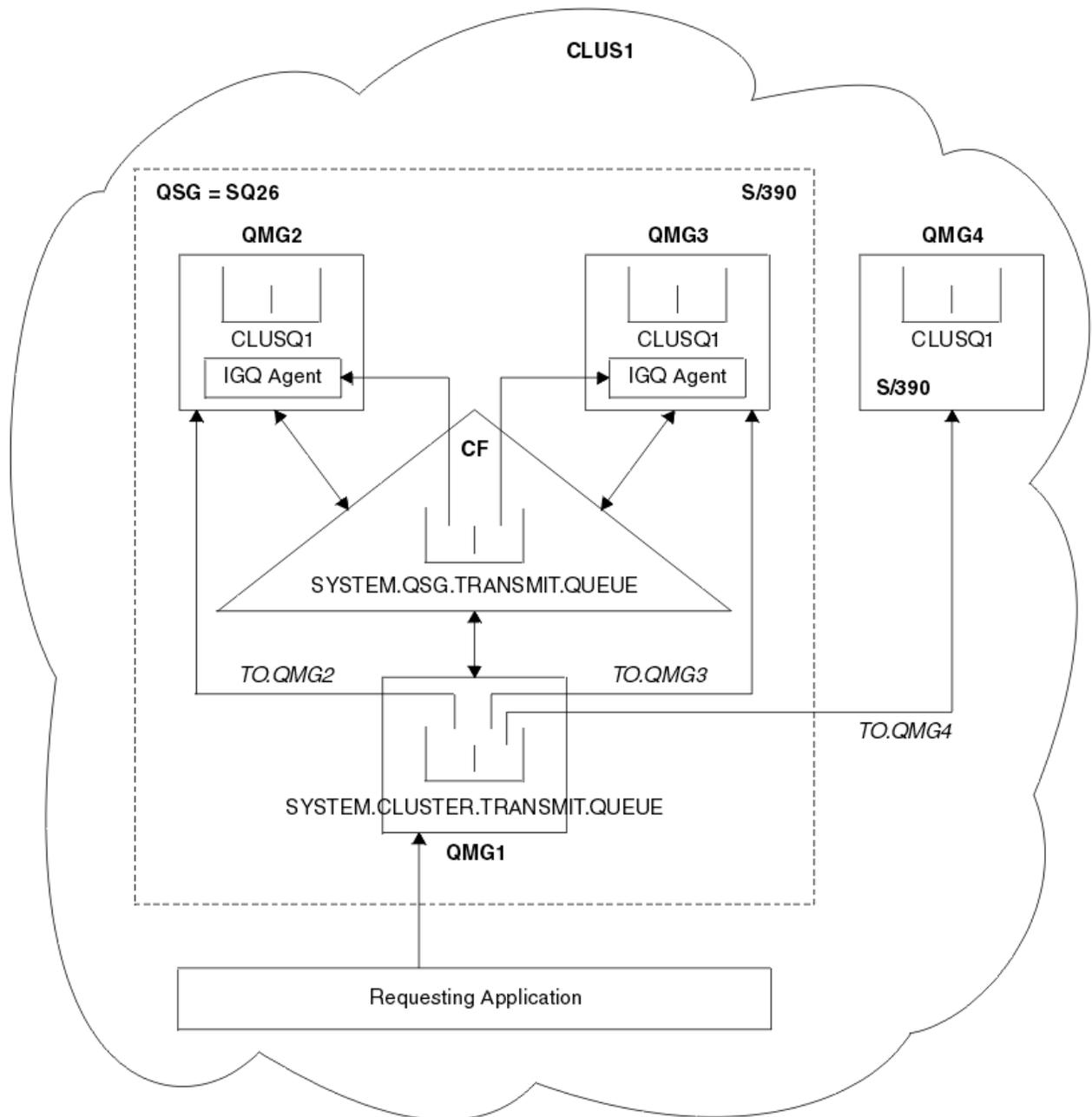


Figure 71. Exemple de mise en cluster avec mise en file d'attente intra-groupe

Le diagramme présente :

- Quatre gestionnaires de files d'attente z/OS QMG1, QMG2, QMG3 et QMG4 configurés dans le cluster CLUS1.
- Les gestionnaires de files d'attente QMG1, QMG2 et QMG3 configurés dans un groupe de partage de files d'attente, SQ26.
- Les agents de mise en file d'attente intra-groupe exécutés sur les gestionnaires de files d'attente QMG2 et QMG3.

- La file d'attente locale SYSTEM.CLUSTER.TRANSMIT.QUEUE définie sur QMG1.

**Remarque :** Pour des raisons de clarté, SYSTEM.CLUSTER.TRANSMIT.QUEUE sur les autres gestionnaires de files d'attente n'apparaît pas.

- SYSTEM.QSG.TRANSMIT.QUEUE définie dans la fonction CF, qui se trouve dans une structure IBM MQ configurée avec l'attribut CFLEVEL (3) RECOVER (YES).
- Les canaux de cluster TO.QMG2 (connectant QMG1 à QMG2), TO.QMG3 (connectant QMG1 à QMG3) et TO.QMG4 (connectant QMG1 à QMG4).
- La file d'attente de cluster CLUSQ1 hébergée sur les gestionnaires de files d'attente QMG2, QMG3 et QMG4.

Supposons que l'application de demande ouvre la file d'attente de cluster à l'aide de l'option MQOO\_BIND\_NOT\_FIXED, de sorte que le gestionnaire de files d'attente cible de la file d'attente de cluster soit sélectionné lors de l'insertion.

Si le gestionnaire de files d'attente cible sélectionné est QMG2 :

- Tous les messages de grande taille insérés par l'application de demande sont :
  - insérés dans la file d'attente SYSTEM.CLUSTER.TRANSMIT.QUEUE sur QMG1, car SYSTEM.QSG.TRANSMIT.QUEUE se trouve dans une structure CFLEVEL(3) et prend par conséquent en charge les messages dont la taille ne dépasse pas 63 Ko ;
  - transférés vers la file d'attente de cluster CLUSQ1 sur QMG2 à l'aide du canal de cluster TO.QMG2.
- Tous les messages de petite taille insérés par l'application de demande sont :
  - insérés dans la file d'attente de transmission partagée SYSTEM.QSG.TRANSMIT.QUEUE ; (Cette file d'attente se trouve dans une structure configurée à l'aide de l'attribut RECOVER(YES) et est utilisée pour les messages de petite taille persistants et non persistants.)
  - extraits par l'agent de mise en file d'attente intra-groupe sur QMG2 ;
  - insérés dans la file d'attente de cluster CLUSQ1 sur QMG2.

Si le gestionnaire de files d'attente cible sélectionné est QMG4 :

- Etant donné que QMG4 ne fait pas partie du groupe de partage de files d'attente SQ26, tous les messages insérés par l'application de demande sont :
  - insérés dans la file d'attente SYSTEM.CLUSTER.TRANSMIT.QUEUE sur QMG1 ;
  - transférés vers la file d'attente de cluster CLUSQ1 sur QMG4 à l'aide du canal de cluster TO.QMG4.

## Remarques

- L'application de demande n'a pas besoin de connaître le mécanisme sous-jacent utilisé pour la distribution des messages.
- Un mécanisme de distribution potentiellement plus rapide est obtenu pour le transfert de petits messages non persistants entre les gestionnaires de files d'attente dans un groupe de partage de files d'attente (même si les mêmes gestionnaires de files d'attente se trouvent dans un cluster).
- Plusieurs chemins sont disponibles pour la distribution de messages (c-à-d, à la fois le chemin de cluster et le chemin de mise en file d'attente intra-groupe).
- Potentiellement plus rapide, le chemin de mise en file d'attente intra-groupe est préféré au chemin de cluster. En fonction des caractéristiques de message, la distribution de message peut être répartie sur les deux chemins. Les messages peuvent donc être distribués hors séquence. Il est à noter que cette distribution est possible sans égard à l'option MQOO\_BIND\_\* spécifiée par l'application. La mise en file d'attente intra-groupe distribue les messages de la même manière que les noeuds de mise en cluster, selon si l'option MQOO\_BIND\_NOT\_FIXED, MQOO\_BIND\_ON\_OPEN, MQOO\_BIND\_ON\_GROUP ou MQOO\_BIND\_AS\_Q\_DEF est spécifiée lors de l'ouverture.
- Lorsqu'un chemin a été sélectionné et que des messages ont été placés dans les files d'attente de transmission, seul le chemin sélectionné est utilisé pour la distribution de messages. Les messages non

traités dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE ne sont pas détournés vers la file d'attente SYSTEM.CLUSTER.TRANSMIT.QUEUE.

## **Mise en cluster, mise en file d'attente intra-groupe et mise en file d'attente répartie**

Il est possible de configurer un gestionnaire de files d'attente qui fait partie d'un cluster et d'un groupe de partage de files d'attente et qui est connecté à un gestionnaire de files d'attente réparties à l'aide d'une paire de canaux émetteur-récepteur.

Cette configuration est une combinaison de la mise en file d'attente répartie avec la mise en file d'attente intra-groupe et de la mise en cluster avec la mise en file d'attente intra-groupe.

La mise en file d'attente intra-groupe est décrite dans la rubrique [«Mise en files d'attente réparties avec mise en files d'attente intra-groupe \(chemins de livraison multiples\)»](#), à la page 233.

La mise en cluster avec la mise en file d'attente intra-groupe est décrite dans la rubrique [«Mise en cluster avec mise en file d'attente intra-groupe \(plusieurs chemins de distribution\)»](#), à la page 235.

## **Messages en file d'attente intra-groupe**

Cette section décrit les messages placés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

### **Structure des messages**

A l'instar de tous les autres messages insérés dans les files d'attente de transmission, les messages insérés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE sont précédés de l'en-tête de file d'attente de transmission (MQXQH).

### **Persistance des messages**

Dans IBM WebSphere MQ 5.3 et les versions ultérieures, les files d'attente partagées prennent en charge les messages persistants et non persistants.

Si le gestionnaire de files d'attente s'arrête lors du traitement de messages non persistants par l'agent de mise en file d'attente intra-groupe ou si ce dernier prend fin de façon anormale en plein milieu du traitement des messages, il se peut que les messages non persistants en cours de traitement soient perdus. Les applications doivent prévoir la reprise des messages non persistants si leur reprise est requise.

Si une demande d'insertion d'un message non persistant, émis par l'agent de mise en file d'attente intra-groupe, échoue inopinément, le message à traiter est perdu.

### **Distribution des messages**

L'agent de mise en file d'attente intra-groupe extrait et distribue tous les messages non persistants hors de la portée d'un point de synchronisation, ainsi que tous les messages objet d'un point de synchronisation. Dans ce cas, l'agent de mise en file d'attente intra-groupe sert de coordinateur de point de synchronisation. Il traite donc les messages non persistants de la même façon dont les messages rapides non persistants sont traités sur un canal de transmission de messages. Voir [Messages rapides non persistants](#).

### **Traitement par lots des messages**

L'agent de mise en file d'attente intra-groupe utilise une taille de lot fixe de 50 messages. Tous les messages persistants extraits dans un lot sont validés à par intervalles de 50 messages. L'agent valide un lot constitué de messages persistants lorsqu'aucun autre message n'est disponible pour extraction dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

### **Taille du message**

La taille maximale d'un message pouvant être inséré dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE correspond à la longueur de message maximale prise en charge pour les files d'attente moins la longueur d'un en-tête de file d'attente de transmission (MQXQH).

### **Persistance de message par défaut et priorité de message par défaut**

Si la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE se trouve dans le chemin de résolution de nom de file d'attente établi au moment de l'ouverture, alors pour les messages insérés à l'aide de la persistance par défaut et de la priorité par défaut (ou à l'aide de la persistance par défaut ou de la

priorité par défaut), les règles normales s'appliquent à la sélection de la file d'attente dont les valeurs de priorité et de persistance par défaut sont utilisées. (Voir la section des [messages IBM MQ](#) pour plus d'informations sur les règles de sélection de file d'attente.)

### Concepts associés

«Messages non livrés/non traités», à la page 239

Cette rubrique décrit ce qui se produit au niveau des messages non livrés et non traités dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

«Messages de rapport - Files d'attente intra-groupes», à la page 239

Cette rubrique décrit les messages de rapport : confirmation à l'arrivée, confirmation à la livraison, rapport d'expiration et rapport d'exception.

### **Messages non livrés/non traités**

Cette rubrique décrit ce qui se produit au niveau des messages non livrés et non traités dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

Si un agent de mise en file d'attente intra-groupe ne peut distribuer un message à la file d'attente de destination, l'agent de mise en file d'attente intra-groupe :

- honore l'option de rapport MQRO\_DISCARD\_MSG (si cela est indiqué comme étant obligatoire dans la zone Options de rapport du MQMD pour le message non livré) et supprime le message non livré ;
- tente de placer le message non livré dans la file d'attente de rebut pour le gestionnaire de files d'attente de destination, si le message n'a pas déjà été supprimé. L'agent de mise en file d'attente intra-groupe fait précéder le message d'un en-tête de file d'attente de rebut (MQDLH).

Si une file d'attente de rebut n'est pas définie ou si un message non livré ne peut pas être placé dans la file d'attente de rebut et si le message non livré est :

- persistant, l'agent de mise en file d'attente intra-groupe annule le lot de messages persistants en cours de traitement, puis passe à l'état de relance. Pour plus d'informations, voir «Propriétés spécifiques à la mise en file d'attente intra-groupe», à la page 240.
- non persistant, l'agent de mise en file d'attente intra-groupe supprime le message et continue de traiter le message suivant.

Si un gestionnaire de files d'attente d'un groupe de partage de files d'attente prend fin avant que l'agent de mise en file d'attente intra-groupe qui lui est associé n'ait eu le temps de traiter tous ses messages, les messages non traités sont conservés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE jusqu'au prochain démarrage du gestionnaire de files d'attente. L'agent de mise en file d'attente intra-groupe extrait ensuite les messages et les distribue aux files d'attente de destination.

Si l'unité de couplage échoue avant que tous les messages de la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE aient été traités, les messages non persistants non traités sont perdus.

IBM recommande que les applications ne placent pas de messages directement dans les files d'attente de transmission. Si une application insère des messages directement dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE, il se peut que l'agent de mise en file d'attente intra-groupe ne puisse pas traiter ces messages et ces derniers sont conservés dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE. Les utilisateurs doivent ensuite utiliser leurs propres méthodes pour traiter ces messages non traités.

### **Messages de rapport - Files d'attente intra-groupes**

Cette rubrique décrit les messages de rapport : confirmation à l'arrivée, confirmation à la livraison, rapport d'expiration et rapport d'exception.

#### **Messages de rapport de confirmation à l'arrivée/confirmation à la livraison**

Les messages de confirmation à l'arrivée et à la livraison sont générés par le gestionnaire de files d'attente lors de l'utilisation de la mise en file d'attente intra-groupe.

#### **Messages de rapport d'expiration**

Les messages de rapport d'expiration sont générés par le gestionnaire de files d'attente.

## Messages de rapport d'exception

En fonction de l'option de rapport MQRO\_EXCEPTION\_\* spécifiée dans la zone *Options de rapport* du descripteur du message non livré, l'agent de mise en file d'attente intra-groupe génère le rapport d'exception requis et le place dans la file d'attente de réponses spécifiée. La mise en file d'attente intra-groupe permet de livrer le rapport d'exception à la file d'attente de réponses de destination.

La persistance du message de rapport est identique à celle du message non livré. Si l'agent de mise en file d'attente intra-groupe ne parvient pas à résoudre le nom de la file d'attente de réponses de destination ou à placer le message de réponse dans une file d'attente de transmission (pour le transférer ultérieurement vers la file d'attente de réponses de destination), il tente d'insérer le rapport d'exception dans la file d'attente de rebut du gestionnaire de files d'attente sur lequel le message de rapport est généré. Au cas où cela ne serait pas possible, si le message non livré est :

- persistant, l'agent de mise en file d'attente intra-groupe supprime le rapport d'exception, annule le lot de messages en cours puis passe à l'état de relance. Pour plus d'informations, voir «[Propriétés spécifiques à la mise en file d'attente intra-groupe](#)», à la page 240.
- non persistant, l'agent de mise en file d'attente intra-groupe supprime le rapport d'exception et continue de traiter le message suivant dans la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE.

## Sécurité de la mise en file d'attente intra-groupe

Cette rubrique décrit les mesures de sécurité relatives à la mise en file d'attente intra-groupe.

Les attributs de gestionnaire de files d'attente IGQAUT (droit de mise en file d'attente intra-groupe) et IGQUSER (ID utilisateur de l'agent de mise en file d'attente intra-groupe) peuvent être définis pour contrôler le niveau de contrôle de sécurité effectué lorsque l'agent de mise en file d'attente intra-groupe ouvre des files d'attente de destination.

### Droit de mise en file d'attente intra-groupe (IGQAUT)

L'attribut IGQAUT peut être paramétré pour indiquer le type de contrôle de sécurité à effectuer et ainsi de déterminer les ID utilisateur à utiliser par l'agent de mise en file d'attente intra-groupe lorsque celui-ci établit le droit d'insertion de messages dans la file d'attente de destination.

L'attribut IGQAUT est similaire à l'attribut PUTAUT qui est disponible sur les définitions de canal.

### Identificateur d'utilisateur de mise en file d'attente intra-groupe (IGQUSER)

L'attribut IGQUSER permet de désigner un ID utilisateur à utiliser par l'agent de mise en file d'attente intra-groupe lorsque celui-ci établit le droit d'insertion de messages dans une file d'attente de destination.

L'attribut IGQUSER est similaire à l'attribut MCAUSER qui est disponible sur les définitions de canal.

## Propriétés spécifiques à la mise en file d'attente intra-groupe

Cette section décrit les propriétés spécifiques de la mise en file d'attente intra-groupe et notamment les fonctionnalités d'invalidation des descripteurs d'objet, de reprise automatique et de relance de l'agent de mise en file d'attente intra-groupe, ainsi que l'agent de mise en file d'attente intra-groupe et la sérialisation.

### Invalidation des descripteurs d'objet (MQRC\_OBJECT\_CHANGED)

Si les attributs d'un objet ont été modifiés à la suite de l'ouverture de l'objet, le gestionnaire de files d'attente invalide le descripteur d'objet à l'aide de MQRC\_OBJECT\_CHANGED lors de sa prochaine utilisation.

La mise en file d'attente intra-groupe introduit les règles suivantes pour l'invalidation du descripteur d'objet :

- Si la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE a été incluse dans le chemin de résolution de nom lors du processus d'ouverture car la mise en file d'attente intra-groupe a été activée (ENABLED) au

moment de l'ouverture alors que la mise en file d'attente intra-groupe est désactivée (DISABLED) au moment de l'insertion, le gestionnaire de files d'attente invalide le descripteur d'objet et fait échouer la demande d'insertion avec MQR\_OBJECT\_CHANGED.

- Si la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE n'a pas été incluse dans le chemin de résolution de nom lors du processus d'ouverture car la mise en file d'attente intra-groupe a été désactivée (DISABLED) au moment de l'ouverture alors que la mise en file d'attente intra-groupe est activée (ENABLED) au moment de l'insertion, le gestionnaire de files d'attente invalide le descripteur d'objet et fait échouer la demande d'insertion avec MQR\_OBJECT\_CHANGED.
- Si la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE a été incluse dans le chemin de résolution de nom lors du processus d'ouverture car la mise en file d'attente intra-groupe a été activée au moment de l'ouverture alors que la définition SYSTEM.QSG.TRANSMIT.QUEUE a été modifiée au moment de l'insertion, le gestionnaire de files d'attente invalide le descripteur d'objet et fait échouer la demande d'insertion avec MQR\_OBJECT\_CHANGED.

## Reprise automatique de l'agent de mise en file d'attente intra-groupe

En cas d'arrêt anormal de l'agent de mise en file d'attente intra-groupe, le message CSQM067E est émis et l'agent redémarre.

## Fonction de relance de l'agent de mise en file d'attente intra-groupe

Si l'agent de mise en file d'attente intra-groupe rencontre un problème lors de l'accès à la file d'attente SYSTEM.QSG.TRANSMIT.QUEUE (car il n'est pas défini, par exemple, ou qu'il est défini à l'aide d'attributs incorrects ou qu'il n'est pas accessible en extraction ou pour une autre raison), l'agent de mise en file d'attente intra-groupe passe à l'état de relance.

L'agent de mise en file d'attente intra-groupe respecte des nombres et intervalles de relances abrégées et longues. Les valeurs de ces nombres et intervalles, qui ne peuvent pas être modifiées, sont les suivantes :

Constante	Valeur
Nombre de relances courtes	10
Intervalle entre relances courtes	60 secondes = 1 min
Nombre de relances longues	999 999 999
Intervalle entre relances longues	1200 secondes = 20 minutes

## Agent de mise en file d'attente intra-groupe et sérialisation

Une tentative de sérialisation par l'agent de mise en file d'attente intra-groupe de l'accès aux files d'attente partagées alors que la reprise homologue est toujours en cours risque d'échouer.

En cas d'incident d'un gestionnaire de files d'attente dans un groupe de partage de files d'attente lorsque l'agent de mise en file d'attente intra-groupe traite des messages non validés d'une ou de plusieurs files d'attente partagées, l'agent de mise en file d'attente intra-groupe s'arrête et la reprise homologue de file d'attente partagée a lieu pour le gestionnaire de files d'attente défaillant. Etant donné que la reprise homologue de file d'attente partagée est une activité asynchrone, elle permet au gestionnaire de files d'attente défaillant et également à l'agent de mise en file d'attente intra-groupe de ce gestionnaire de files d'attente de redémarrer avant la fin de la reprise homologue de file d'attente partagée, ce qui, à son tour, permet aux messages non validés d'être traités hors séquence et avant les messages restant à récupérer. Pour garantir que les messages ne sont pas traités hors séquence, l'agent de mise en file d'attente intra-groupe sérialise l'accès aux files d'attente partagées en émettant l'appel d'API MQCONN.

Une tentative de sérialisation par l'agent de mise en file d'attente intra-groupe de l'accès aux files d'attente partagées alors que la reprise homologue est toujours en cours risque d'échouer. Un message d'erreur est émis et l'agent de mise en file d'attente intra-groupe passe à l'état de relance. Une fois la

reprise homologue du gestionnaire de files d'attente terminée, par exemple lors de la prochaine relance, l'agent de mise en file d'attente intra-groupe peut démarrer.

## z/OS Gestion du stockage sur z/OS

IBM MQ for z/OS requiert des structures de données permanentes et temporaires et utilise des ensembles de pages et des tampons mémoire pour archiver ces données. Ces rubriques contiennent d'autres informations sur le mode d'utilisation de ces ensembles de pages et de ces tampons par IBM MQ.

### Concepts associés

«Ensembles de pages pour IBM MQ for z/OS», à la page 242

Consultez cette rubrique pour comprendre comment IBM MQ for z/OS utilise les ensembles de pages pour stocker les messages.

«Classes d'archivage pour IBM MQ for z/OS», à la page 243

Une classe d'archivage est un concept IBM MQ for z/OS qui permet au gestionnaire de files d'attente d'associer des files d'attente à des ensembles de pages. Vous pouvez utiliser des classes d'archivage pour contrôler quels fichiers sont utilisés par quelles files d'attente.

«Mémoires tampons et pools de mémoires tampons pour IBM MQ for z/OS», à la page 244

IBM MQ for z/OS utilise des tampons et des pools de mémoire tampon pour mettre temporairement les données en mémoire cache. Consultez cette rubrique pour comprendre comment les tampons sont organisés et utilisés.

### Référence associée

«Où trouver des informations supplémentaires sur la gestion du stockage pour IBM MQ for z/OS», à la page 246

Utilisez cette rubrique comme référence pour rechercher d'autres informations sur la gestion de l'archivage pour IBM MQ for z/OS.

## z/OS Ensembles de pages pour IBM MQ for z/OS

Consultez cette rubrique pour comprendre comment IBM MQ for z/OS utilise les ensembles de pages pour stocker les messages.

Un *ensemble de pages* est un fichier linéaire VSAM spécialement formaté pour être utilisé par IBM MQ. Les ensembles de pages permettent de stocker la plupart des messages et des définitions d'objet.

Les définitions globales, stockées dans un référentiel partagé sur Db2, et les messages, stockés dans des files d'attente partagées, sont les exceptions à cette règle. Ils ne sont pas stockés dans les ensembles de pages du gestionnaire de files d'attente. Pour plus d'informations sur les files d'attente partagées, voir «Files d'attente partagées et groupes de partage de files d'attente», à la page 176, et pour plus d'informations sur les définitions globales, voir [Définitions privées et globales](#).

Les ensembles de pages IBM MQ peuvent avoir une taille pouvant atteindre 64 Go. Chaque ensemble de pages est identifié par un identificateur d'ensemble de pages (PSID) qui est un entier compris entre 00 et 99. Chaque gestionnaire de files d'attente doit être associé à ses propres ensembles de pages.

IBM MQ utilise l'ensemble de pages zéro (PSID=00) pour stocker les définitions d'objet et d'autres informations importantes pour le gestionnaire de files d'attente. Pour qu'IBM MQ fonctionne normalement, il est essentiel que l'ensemble de pages zéro ne soit pas plein. Vous ne devez donc pas l'utiliser pour y stocker des messages.

Pour améliorer les performances de votre système, vous devez également séparer les messages de courte durée des messages de longue durée en les plaçant dans des ensembles de pages distincts.

Vous devez formater les ensembles de pages à l'aide de l'utilitaire `FORMAT` fourni par IBM MQ. Voir [Formatage des ensembles de pages \(FORMAT\)](#). Les ensembles de pages doivent également être définis dans le sous-système IBM MQ.

IBM MQ for z/OS peut être configuré de sorte à étendre de façon dynamique un ensemble de pages qui est plein. IBM MQ continue de développer l'ensemble de pages si nécessaire jusqu'à atteindre 123

extensions logiques, si l'espace de stockage sur disque disponible est suffisant. Ces extensions peuvent être réparties sur différents volumes si le fichier linéaire est défini de cette manière. Toutefois, IBM MQ ne peut pas étendre les ensembles de pages au-delà de 64 Go.

Vous ne pouvez pas utiliser les ensembles de pages d'un gestionnaire de files d'attente IBM MQ dans un autre gestionnaire de files d'attente différent IBM MQ ni modifier le nom de ce gestionnaire. Si vous souhaitez transférer les données d'un gestionnaire de files d'attente à un autre, vous devez télécharger tous les objets et messages du premier gestionnaire de files d'attente et les recharger dans l'autre.

Il est impossible d'utiliser des ensembles de pages dont la taille est supérieure à 4 Go dans un gestionnaire de files d'attente exécutant une édition antérieure à la version 6. Lors de la migration, si vous pensez devoir revenir à une édition précédente du code, procédez comme suit :

- N'augmentez pas la taille de l'ensemble de pages 0 à une taille supérieure à 4 Go.
- Les autres ensembles de pages dont la taille est supérieure à 4 Go resteront hors ligne lors du redémarrage d'un gestionnaire de files d'attente avec une édition précédente.

Pour plus d'informations sur la migration des ensembles de pages existants pouvant être étendus au-delà de 4 Go, voir [Définition d'un ensemble de pages supérieur à 4 Go](#).

Un administrateur peut ajouter ou supprimer des ensembles de pages de manière dynamique dans un gestionnaire de files d'attente en cours d'exécution (à l'exception du gestionnaire de files d'attente zéro). La commande DEFINE PSID peut être exécutée une fois que le gestionnaire de files d'attente a redémarré uniquement si elle contient le mot clé DSN.

## Classes d'archivage pour IBM MQ for z/OS

Une classe d'archivage est un concept IBM MQ for z/OS qui permet au gestionnaire de files d'attente d'associer des files d'attente à des ensembles de pages. Vous pouvez utiliser des classes d'archivage pour contrôler quels fichiers sont utilisés par quelles files d'attente.

### Présentation des classes d'archivage

Une *classe d'archivage* mappe une ou plusieurs files d'attente vers un ensemble de pages. Cela signifie que les messages de cette file d'attente sont stockés sur cet ensemble de pages.

Les classes d'archivage vous permettent de contrôler l'emplacement du stockage des données de messages non partagées à des fins d'administration, de gestion de l'espace de fichiers et de la charge ou d'isolement des applications. Vous pouvez également utiliser ces classes pour définir le groupe XCF et le nom de membre d'une région IMS si vous utilisez la passerelle IMS (décrite dans [«IBM MQ et IMS»](#), à la page 306).

Les files d'attente partagées n'utilisent pas les classes d'archivage pour obtenir un mappage de fichiers car les messages s'y trouvant ne sont pas stockés sur les ensembles de pages.

### Fonctionnement des classes d'archivage

- Vous définissez une classe d'archivage à l'aide de la commande DEFINE STGCLASS et vous spécifiez un identificateur d'ensemble de pages (PSID).
- Lorsque vous définissez une file d'attente, vous spécifiez la classe d'archivage de l'attribut STGCLASS.

Dans l'exemple suivant, la file d'attente locale QE5 est mappée vers l'ensemble de pages 21 via la classe d'archivage ARC2.

```
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE QLOCAL(QE5) STGCLASS(ARC2)
```

Cela signifie que les messages placés dans la file d'attente QE5 sont stockés dans l'ensemble de pages 21 (s'ils restent dans la file d'attente suffisamment longtemps pour être écrits sur le disque).

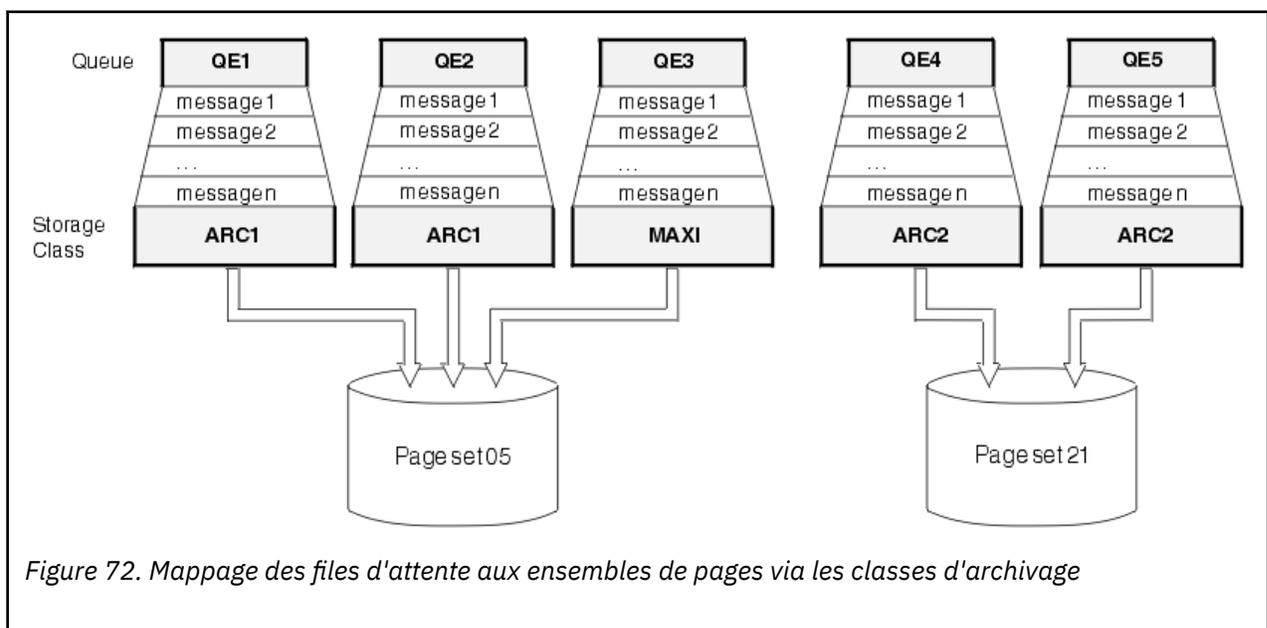
Plusieurs files d'attente peuvent utiliser la même classe d'archivage et vous pouvez définir autant de classes d'archivage que vous le souhaitez. Vous pouvez par exemple développer l'exemple précédent et y ajouter d'autres classes d'archivage et définitions de files d'attente, comme suit :

```

DEFINE STGCLASS(ARC1) PSID(05)
DEFINE STGCLASS(ARC2) PSID(21)
DEFINE STGCLASS(MAXI) PSID(05)
DEFINE QLOCAL(QE1) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE2) STGCLASS(ARC1) ...
DEFINE QLOCAL(QE3) STGCLASS(MAXI) ...
DEFINE QLOCAL(QE4) STGCLASS(ARC2) ...
DEFINE QLOCAL(QE5) STGCLASS(ARC2) ...

```

Dans la Figure 72, à la page 244, les classes d'archivage ARC1 et MAXI sont associées à l'ensemble de pages 05. Les files d'attente QE1, QE2 et QE3 sont mappées vers l'ensemble de pages 05. De même, la classe d'archivage ARC2 associe les files d'attente QE4 et QE5 à l'ensemble de pages 21.



Si vous définissez une file d'attente sans spécifier de classe d'archivage, IBM MQ utilise une classe d'archivage par défaut.

Si un message est placé sur une file d'attente qui nomme une classe d'archivage inexistante, l'application reçoit une erreur. Vous devez modifier la définition de file d'attente en lui donnant un nom de classe d'archivage existant ou créer la classe d'archivage nommée par la file d'attente.

Vous pouvez modifier une classe d'archivage uniquement lorsque :

- Toutes les files d'attente utilisant cette classe d'archivage sont vides et ne sont associées à aucune activité non validée.
- Toutes les files d'attente utilisant cette classe d'archivage sont fermées.

## z/OS Mémoires tampons et pools de mémoires tampons pour IBM MQ for z/OS

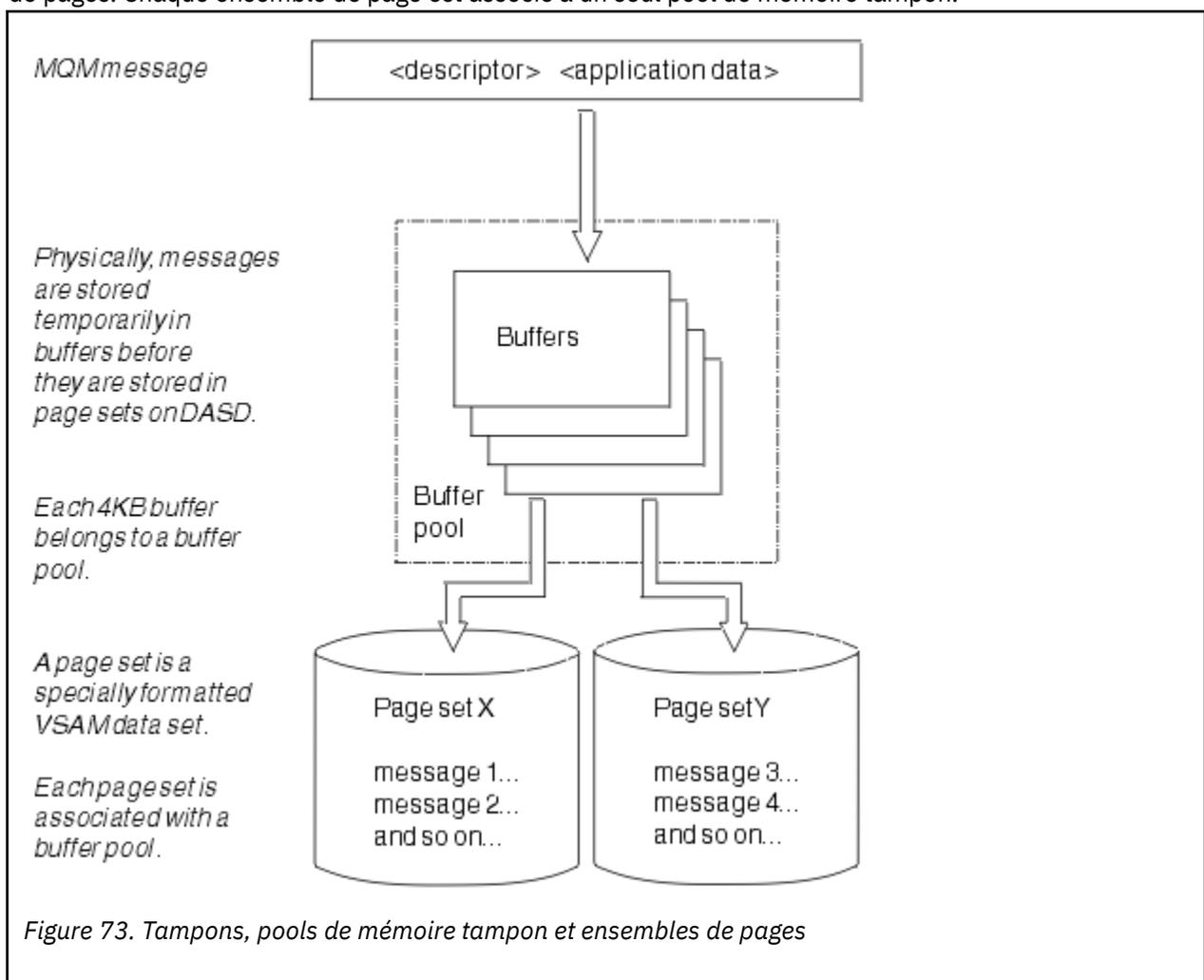
IBM MQ for z/OS utilise des tampons et des pools de mémoire tampon pour mettre temporairement les données en mémoire cache. Consultez cette rubrique pour comprendre comment les tampons sont organisés et utilisés.

Pour une plus grande efficacité, IBM MQ utilise une forme de mise en cache selon laquelle les messages (et les définitions d'objet) sont stockés temporairement dans des tampons avant d'être stockés dans les ensembles de pages d'un disque. Les messages de courte durée, c'est-à-dire les messages qui sont extraits d'une file d'attente peu après avoir été reçus, ne sont probablement jamais stockés dans les tampons. Cette activité de mise en cache est contrôlée par un gestionnaire de mémoire tampon, qui est un composant de IBM MQ.

Les tampons sont organisés en *pools de mémoire tampon*. Vous pouvez définir jusqu'à 100 pools de mémoire tampon (0 à 99) pour chaque gestionnaire de files d'attente.

Il est recommandé d'utiliser le nombre minimal de pools de mémoire tampon correspondant à la séparation d'objet et de type de message décrite dans Figure 73, à la page 245, ainsi que de respecter toutes les exigences d'isolement de données de votre application. Chaque tampon a une longueur de 4 ko. Les pools de mémoire utilisent le stockage 31 bits par défaut. Avec ce mode, le nombre maximal de mémoires tampons est déterminé par la quantité de stockage 31 bits disponible dans l'espace adresse du gestionnaire de files d'attente. N'en utilisez pas plus de 70% pour les mémoires tampons. L'allocation de mémoire du pool de mémoire tampon peut également être effectuée à partir d'une mémoire 64 bits (utilisez l'attribut LOCATION de la commande **DEFINE BUFFPOOL**). L'utilisation de LOCATION(ABOVE) pour que le stockage 64 bits soit utilisé présente deux avantages. Premièrement, le stockage disponible 64 bits est bien plus important, ce qui permet d'avoir des pools de mémoire tampon plus volumineux. Ensuite, le stockage 31 bits est rendu disponible pour les autres fonctions. Généralement, plus vous avez de mémoires tampons, plus la mise en mémoire tampon est efficace et meilleures sont les performances d'IBM MQ.

La Figure 73, à la page 245 présente la relation entre les messages, les tampons, les pools de mémoire tampon et les ensembles de pages. Un pool de mémoire tampon est associé à un ou plusieurs ensembles de pages. Chaque ensemble de page est associé à un seul pool de mémoire tampon.



Vous pouvez exécuter des commandes de manière dynamique pour modifier la taille et l'emplacement du pool de mémoire tampon à l'aide de la commande **ALTER BUFFPOOL**. Les ensembles de pages peuvent être ajoutés dynamiquement à l'aide de la commande **DEFINE PSID** ou supprimés à l'aide de la commande **DELETE PSID**.

Si un pool de mémoire tampon est trop petit, IBM MQ émet le message CSQP020E. Vous pouvez alors ajouter dynamiquement d'autres tampons au pool concerné (pour cela, vous devrez peut-être supprimer des tampons dans d'autres pools).

Vous spécifiez le nombre de tampons dans un pool à l'aide de la commande **DEFINE BUFFPOOL** et vous pouvez redimensionner les pools de mémoire tampon de manière dynamique à l'aide de la commande **ALTER BUFFPOOL**. Pour déterminer de manière dynamique le nombre de tampons en cours dans un pool, vous affichez un ensemble de pages qui utilise le pool de mémoire tampon à l'aide de la commande **DISPLAY USAGE**.

Pour des raisons liées aux performances, ne placez pas les messages et les définitions d'objet dans le même pool de mémoire tampon. Utilisez un pool de mémoire tampon (par exemple le pool zéro) exclusivement pour l'ensemble de pages zéro, dans lequel les définitions d'objet sont conservées. De même, conservez les messages de courte durée et les messages de longue durée dans des pools distincts, donc dans des ensembles de pages distincts, ainsi que dans des files d'attente distinctes.

Il est impossible d'utiliser la commande **DEFINE BUFFPOOL** après le redémarrage pour créer un pool de mémoire tampon. En revanche, si une commande **DEFINE PSID** utilise le mot clé DSN, elle peut explicitement identifier un pool de mémoire tampon actuellement non défini. Ce nouveau pool est alors créé.

## z/OS Où trouver des informations supplémentaires sur la gestion du stockage pour IBM MQ for z/OS

Utilisez cette rubrique comme référence pour rechercher d'autres informations sur la gestion de l'archivage pour IBM MQ for z/OS.

Vous pouvez obtenir d'autres informations sur les rubriques de cette section à partir des sources suivantes :

Tableau 20. Où trouver des informations supplémentaires sur la gestion du stockage ?	
Topic	Section
Quantité de mémoire requise	<a href="#">Planification de l'archivage et exigences de performance sur z/OS</a>
Taille requise des ensembles de pages et des pools de mémoire tampon	<a href="#">Planification de vos ensembles de pages et de vos pools de mémoire tampon</a>
Gestion des ensembles de pages	<a href="#">Gestion des ensembles de pages</a>
Commandes MQSC	<a href="#">Les commandes MQSC</a>

## z/OS Journalisation dans IBM MQ for z/OS

IBM MQ gère des *journaux* des modifications des données et des événements importants. Ces journaux peuvent, si nécessaire, servir à restaurer l'état précédent de certaines données.

Le *fichier d'amorce* stocke les informations relatives aux fichiers contenant les journaux.

Le journal ne contient aucune information relative aux statistiques, aux traces ou à l'évaluation des performances. Pour plus de détails sur les informations statistiques et de surveillance collectées par IBM MQ, voir [Surveillance et statistiques](#).

Pour plus d'informations sur la journalisation, voir les rubriques suivantes :

- «Fichiers journaux dans IBM MQ for z/OS», à la page 247
- «Structuration du journal», à la page 251
- «Comment les journaux d'IBM MQ for z/OS sont-ils écrits ?», à la page 252
- «Adresse relative en octets de journal de plus grande taille», à la page 255
- «Le fichier d'amorce», à la page 257

### Tâches associées

Planification de votre environnement de journalisation

Définition des journaux à l'aide du module des paramètres système

 Administration de z/OS

### Référence associée

 Messages pour IBM MQ for z/OS

Sources à partir desquelles vous pouvez émettre des commandes MQSC sous z/OS

## Fichiers journaux dans IBM MQ for z/OS

Les fichiers journaux contiennent les informations nécessaires à la récupération des transactions. Vous pouvez archiver les jeux de données de journal actifs afin de conserver les données de journalisation pendant une longue période.

### Qu'est-ce qu'un fichier journal ?

IBM MQ enregistre tous les événements importants au fur et à mesure qu'ils se produisent dans un *journal actif*. Ce journal contient les informations nécessaires pour récupérer :

- les messages persistants
- les objets IBM MQ, comme les files d'attente
- Le gestionnaire de files d'attente IBM MQ

Le journal actif contient un ensemble de fichiers (310 au maximum) utilisés de manière cyclique.

Vous pouvez activer l'archivage des journaux de telle sorte que lorsqu'un journal actif est plein, son contenu est copié dans un fichier d'archivage. L'utilisation de la fonction d'archivage vous permet de conserver des données pendant une longue période. Si vous n'utilisez pas cette fonction, les journaux exécutent une boucle et les données antérieures sont écrasées. Pour récupérer un ensemble de pages ou les données d'une structure d'unité de couplage, vous avez besoin des données de journalisation datant du moment où la sauvegarde de l'ensemble de pages ou de la structure a été effectuée. Vous pouvez créer les journaux d'archivage sur un disque ou sur une bande.

### Archivage

La taille du journal actif étant fixe, IBM MQ copie à intervalles réguliers le contenu de chaque fichier d'archivage dans un *journal d'archivage*, qui est normalement un fichier stocké sur une unité de stockage à accès direct ou sur une bande magnétique. En cas de défaillance du sous-système ou d'une transaction, IBM MQ utilise le journal actif et, si nécessaire, le fichier d'archivage pour la récupération.

Le journal d'archivage peut contenir 1000 fichiers séquentiels au maximum. Vous pouvez cataloguer chaque fichier à l'aide de la fonction de catalogue intégrée de z/OS.

L'archivage est un composant essentiel de la récupération IBM MQ. Dans le cas d'une unité de récupération à exécution longue, les enregistrements de journal de cette unité peuvent se trouver dans le journal d'archivage. Dans ce cas, les données du journal d'archivage sont nécessaires à la reprise. Toutefois, si la fonction d'archivage est désactivée, le journal actif contenant les nouveaux enregistrements exécute une boucle et les enregistrements plus anciens sont écrasés. Cela signifie qu'IBM MQ risque de ne pas pouvoir annuler l'unité de récupération et que des messages risquent de se perdre. Le gestionnaire de files d'attente s'arrête alors de manière anormale.

**Ne désactivez donc jamais l'archivage** dans un environnement de production. Si vous désactivez cette fonction, vous risquez de perdre des données après une défaillance du système ou d'une transaction. N'envisagez de désactiver la fonction d'archivage que si vous travaillez dans un environnement de test. Si cette désactivation est nécessaire, utilisez la macro CSQ6LOGP décrite à la section [Utilisation de CSQ6LOGP](#).

Pour prévenir d'éventuels problèmes liés aux unités d'oeuvre à exécution longue non planifiées, IBM MQ émet un message (CSQJ160I ou CSQJ161I) si une unité d'oeuvre à exécution longue est détectée lors du traitement du déchargement du journal actif.

## Double consignation

Avec la double consignation, chaque enregistrement de journal est consigné dans deux jeux de données de journal actifs afin de réduire les risques liés à une éventuelle perte de données lors du redémarrage.

Vous pouvez configurer IBM MQ de sorte à s'exécuter en mode *consignation simple* ou *double consignation*. Avec la consignation simple, les enregistrements de journal sont écrits une fois dans un jeu de données de journal actif. Chaque jeu de données de journal actif est un fichier linéaire de domaine unique VSAM. Avec la double consignation, chaque enregistrement de journal est consigné dans deux jeux de données de journal actifs différents. Ce mode de consignation minimise les risques liés à une éventuelle perte de données lors du redémarrage.

## Dérivation de journal

La dérivation permet d'écrire les enregistrements de certaines unités d'oeuvre dans une partie postérieure du fichier journal. Cette opération permet de réduire la quantité de données à lire lors du redémarrage ou de l'annulation du gestionnaire de files d'attente pour les unités d'oeuvre à exécution longue ou en attente de validation à long terme.

Lorsqu'une unité d'oeuvre est considérée comme longue, une représentation de chaque enregistrement de journal est consignée plus loin dans le journal. Cette technique est appelée *dérivation*. Une fois que la totalité de l'unité d'oeuvre a été traitée, l'unité d'oeuvre est à l'état *dérivé*. Les activités d'annulation ou de redémarrage associées à l'unité d'oeuvre dérivée peuvent utiliser les enregistrements de journal dérivés plutôt que les enregistrements de l'unité d'origine.

La détection d'une unité d'oeuvre à exécution longue est une fonction du traitement du point de contrôle. Lors du point de contrôle, chaque unité d'oeuvre active fait l'objet d'une vérification permettant de définir si elle doit être dérivée. Si cette unité d'oeuvre est passée par deux points de contrôle depuis sa création ou depuis sa dernière dérivation, elle peut faire l'objet d'une dérivation. Cela signifie qu'une même unité d'oeuvre peut être dérivée plusieurs fois. Il s'agit alors d'une unité d'oeuvre *dérivée plusieurs fois*.

Une unité d'oeuvre est dérivée tous les trois points de contrôle. Toutefois, le point de contrôle est effectué en mode asynchrone pour la commutation de journal (ou l'écriture de l'enregistrement de journal qui a entraîné le dépassement de LOGLOAD).

Un seul point de contrôle est effectué à la fois ; par conséquent, il peut y avoir plusieurs commutations de journal avant la fin de l'exécution d'un point de contrôle.

En d'autres termes, si le nombre de journaux actifs n'est pas suffisant ou si les journaux actifs sont trop petits, il se peut que la dérivation d'une unité d'oeuvre volumineuse ne soit pas effectuée avant que tous les journaux ne soient remplis.

Le message [CSQR027I](#) est généré si la dérivation ne peut pas avoir lieu.

Si l'archivage des journaux est désactivé, l'erreur ABEND 5C6 avec le motif 00D1032A est émise si vous tentez d'annuler l'unité d'oeuvre pour laquelle la dérivation a échoué. Pour éviter ce problème, utilisez OFFLOAD=YES.

La dérivation des journaux est toujours active. Elle s'exécute dans tous les cas, que l'archivage des journaux soit activé ou non.

**Remarque :** Bien que tous les enregistrements de journal d'une unité d'oeuvre soient dérivés, seule la partie nécessaire à l'annulation est dérivée (et non la totalité du contenu de chaque enregistrement). Cela signifie que la quantité de données consignées est réduite à son strict minimum et que les enregistrements dérivés ne peuvent pas être utilisés en cas d'erreur liée à l'ensemble de pages. Une unité d'oeuvre à exécution longue est une unité qui s'exécute depuis que trois points de contrôle du gestionnaire de files d'attente ont été exécutés au moins.

Pour plus d'informations sur la dérivation des journaux, voir [Gestion des journaux](#).

## Compression de journal

Vous pouvez configurer IBM MQ for z/OS afin de compresser et décompresser les enregistrements de journal lors de leur consignation et de leur lecture dans le jeu de données de journal.

La compression de journal permet de réduire le nombre de données consignées dans le journal pour les messages persistants se trouvant sur les files d'attente privées. Le niveau de compression atteint dépend du type de données contenu dans les messages. Par exemple, l'algorithme RLE procède en compactant des instances répétées d'octets, ce qui peut donner de bons résultats pour les données structurées ou orientées enregistrement.



**Avertissement :** Les messages persistants qui sont placés dans une file d'attente partagée ne sont pas soumis à la compression de journal.

Vous pouvez utiliser les zones de la section Gestionnaire de journaux des enregistrements SMF (System Management Facility 115) pour surveiller le niveau de compression des données. Pour plus d'informations sur SMF, voir [Utilisation de la fonction de gestion du système](#) et [Messages de comptabilité et de statistiques](#).

La compression de journal permet d'optimiser l'utilisation du processeur du système. Vous devez envisager d'utiliser cette fonction uniquement si la capacité de traitement de votre gestionnaire de files d'attente est limitée par la bande passante d'entrée-sortie consignante les données dans les jeux de données de journal ou par l'espace disque nécessaire pour contenir ces données. Si vous utilisez les files d'attente partagées, vous pouvez contourner les limites liées à la bande passante en ajoutant d'autres gestionnaires de files d'attente au groupe de partage et en répartissant la charge de travail sur un plus grand nombre de gestionnaires de files d'attente.

L'option de compression de journal peut être activée et désactivée selon vos besoins, sans qu'il soit nécessaire d'arrêter et de redémarrer le gestionnaire de files d'attente. Quel que soit le paramètre de compression en cours, le gestionnaire de files d'attente peut lire tous les enregistrements de journal compressés.

Le gestionnaire de files d'attente prend en charge trois paramètres de compression de journal.

### AUCUN

La compression des données de journal n'est pas utilisée. Il s'agit de la valeur par défaut.

### RLE

La compression des données de journal est exécutée par l'algorithme RLE.

### ANY

Activez le gestionnaire de files d'attente pour sélectionner l'algorithme de compression permettant d'atteindre le degré le plus élevé de compression. Cette option active la compression RLE.

Vous pouvez contrôler la compression des enregistrements de journal à l'aide des outils suivants :

- Les commandes SET et DISPLAY LOG dans MQSC. Voir [SET LOG](#) et [DISPLAY LOG](#).
- Les fonctions Set Log et Inquire Log de l'interface PCF. Voir [Set log](#) et [Inquire log](#).
- La macro CSQ6LOGP du module des paramètres système. Voir [Utilisation de CSQ6LOGP](#).

Outre l'utilitaire d'impression des journaux, CSQ1LOGP prend en charge le développement des enregistrements de journal compressés.

## Données de journalisation

Le journal peut contenir 18 millions de millions de millions ( $1.8 \cdot 10^{19}$ ) d'octets au maximum. Chaque octet peut être adressé par son décalage par rapport au début du journal. Ce décalage est appelé *adresse relative en octets*.

Cette adresse est référencée par une zone de 6 ou 8 octets, soit une plage adressable totale de  $2^{48}$  octets ou  $2^{64}$  octets, selon que des adresses relatives en octets de 6 ou 8 octets de journal sont utilisées.

Toutefois, lorsqu'IBM MQ détecte que la plage utilisée dépasse F00000000000 (en cas d'utilisation d'adresses relatives en octets de 6 octets) ou FFFF800000000000 (en cas d'utilisation d'adresses relatives en octets de 8 octets), les messages [CSQI045](#), [CSQI046](#), [CSQI047](#) et [CSQJ032](#) sont émis pour vous avertir que l'adresse relative en octets doit être réinitialisée.

Si la valeur de l'adresse relative en octets atteint FFF800000000 (en cas d'utilisation d'adresses relatives en octets de 6 octets) ou FFFFFFFC0000000000 (en cas d'utilisation d'adresses relatives en octets de 8 octets), le gestionnaire de files d'attente s'arrête avec le code anomalie [00D10257](#).

Une fois que les messages d'avertissement relatifs à la plage des journaux utilisée ont été émis, vous devez planifier une indisponibilité au cours de laquelle le gestionnaire de files d'attente peut être converti pour utiliser des adresses relatives en octets de journal de 8 octets ou le journal peut être réinitialisé. La procédure de réinitialisation du journal est décrite dans [Réinitialisation du journal du gestionnaire de files d'attente](#).

Si votre gestionnaire de files d'attente utilise des adresses relatives en octets de journal de 6 octets, envisagez de convertir le gestionnaire de files d'attente afin qu'il utilise des adresses relatives en octets de journal de 8 octets au lieu de réinitialiser le journal du gestionnaire de files d'attente en suivant la procédure décrite dans [Implémentation de l'adresse relative en octets de journal de plus grande taille](#).

Le journal est constitué d'*enregistrements de journal*, chaque enregistrement étant un ensemble de données traité comme une unité unique. Un enregistrement de journal est identifié par l'adresse relative en octets du premier octet de son en-tête ou par son numéro de séquence d'enregistrement de journal. Cette adresse ou ce numéro identifient de manière unique un enregistrement commençant à un point donné du journal.

Selon que votre environnement contient des groupes de partage de files d'attente ou non, vous utiliserez les adresses ou les numéros. Dans un environnement de partage de files d'attente, vous ne pouvez pas utiliser les adresses relatives en octets car plusieurs gestionnaires de files d'attente peuvent mettre à jour la même file d'attente en même temps et car chacun a son propre journal. Pour résoudre ce problème, le numéro de séquence de l'enregistrement de journal est dérivé d'une valeur d'horodatage et il ne représente pas nécessairement le déplacement physique de l'enregistrement dans le journal.

Chaque enregistrement de journal est associé à un en-tête indiquant son type, le sous-composant d'IBM MQ qui est à l'origine de l'enregistrement et, pour les enregistrements d'unité de récupération, un identificateur d'unité de récupération.

Il existe quatre types d'enregistrement de journal, qui sont décrits dans les rubriques suivantes :

- [Enregistrements de journal des unités de récupération](#)
- [Enregistrements des points de contrôle](#)
- [Enregistrements de contrôle des ensembles de pages](#)
- [Enregistrements de sauvegarde des structures d'unité de couplage](#)

## Enregistrements de journal des unités de récupération

La plupart des enregistrements de journal décrivent les modifications apportées aux files d'attente IBM MQ. Ces modifications sont apportées dans les unités de récupération.

IBM MQ utilise des techniques de journalisation spéciales impliquant des opérations d'*annulation/reprise* et de *compensation des enregistrements de journal* afin de réduire la durée des redémarrages et d'améliorer la disponibilité du système.

L'utilisation de cette technique a pour conséquence de limiter la durée du redémarrage. Si une erreur se produit lors du redémarrage et que le gestionnaire de files d'attente doit être redémarré une seconde fois, il n'est pas nécessaire, lors du second redémarrage, de relancer l'activité de reprise exécutée jusqu'au point de défaillance qui s'est produit lors du premier redémarrage. Cela signifie que plusieurs redémarrages successifs ne demandent pas plus de temps.

## Enregistrements des points de contrôle

En temps normal, IBM MQ effectue des points de contrôle périodiques afin de réduire la durée du redémarrage. Cette opération se déroule comme suit :

- Lorsqu'un nombre prédéfini d'enregistrements de journal a été consigné. Ce nombre est défini par l'opérande de fréquence du point de contrôle LOGLOAD de la macro des paramètres système CSQ6SYSP décrite à la section [Utilisation de CSQ6SYSP](#).
- A la fin d'un redémarrage ayant abouti.
- Lors d'un arrêt normal.
- Lorsqu'IBM MQ passe au jeu de données de journal actif suivant dans le cycle.

Lorsque le point de contrôle est relevé, IBM MQ émet en interne la commande DISPLAY CONN (décrite dans [DISPLAY CONN](#)). Une liste des connexions en attente de validation est alors consignée dans le journal de la console z/OS.

## Enregistrements de contrôle des ensembles de pages

Ils enregistrent les ensembles de pages et les pools de mémoire tampon connus du gestionnaire de files d'attente IBM MQ lors de chaque point de contrôle, ainsi que les informations relatives aux pages du journal nécessaires pour procéder à la récupération des supports de l'ensemble de pages au moment du point de contrôle.

Certaines modifications dynamiques apportées aux ensembles de pages et aux pools de mémoire tampon sont également consignées en tant qu'enregistrements de contrôle des ensembles de pages. Ces modifications peuvent ainsi être récupérées et automatiquement rétablies lors du redémarrage suivant du gestionnaire de files d'attente.

## Enregistrements de sauvegarde des structures d'unité de couplage

Ces enregistrements contiennent les données lues dans une structure de liste d'unité de couplage en réponse à une commande BACKUP CFSTRUCT. Dans le cas peu probable d'une défaillance liée à une structure d'unité de couplage, ces enregistrements sont utilisés, ainsi que les enregistrements de reprise, par la commande RECOVER CFSTRUCT pour procéder à la récupération des supports de la structure d'unité de couplage jusqu'au point de défaillance.

### Tâches associées

[Implémentation de l'adresse relative en octets de journal de plus grande taille](#)

## Structuration du journal

Consultez cette rubrique pour comprendre la terminologie utilisée pour décrire les enregistrements de journal.

Chaque jeu de données de journal actif doit être un fichier linéaire VSAM. L'unité de sortie physique écrite dans le jeu de données de journal actif est un intervalle de contrôle de 4 ko. Chaque intervalle de contrôle contient un enregistrement VSAM.

## Enregistrements des journaux physique et logique

Un intervalle de contrôle VSAM est un enregistrement *physique*. Les informations consignées à une heure donnée forment un enregistrement *logique* dont la longueur varie indépendamment de l'espace disponible dans l'intervalle de contrôle. Un enregistrement physique peut donc contenir :

- Plusieurs enregistrements logiques
- Un ou plusieurs enregistrements logiques et une partie d'un autre enregistrement logique
- Uniquement une partie d'un enregistrement logique

Le terme *enregistrement de journal* fait référence à l'enregistrement *logique*, quel que soit le nombre d'enregistrements *physiques* nécessaire à son stockage.

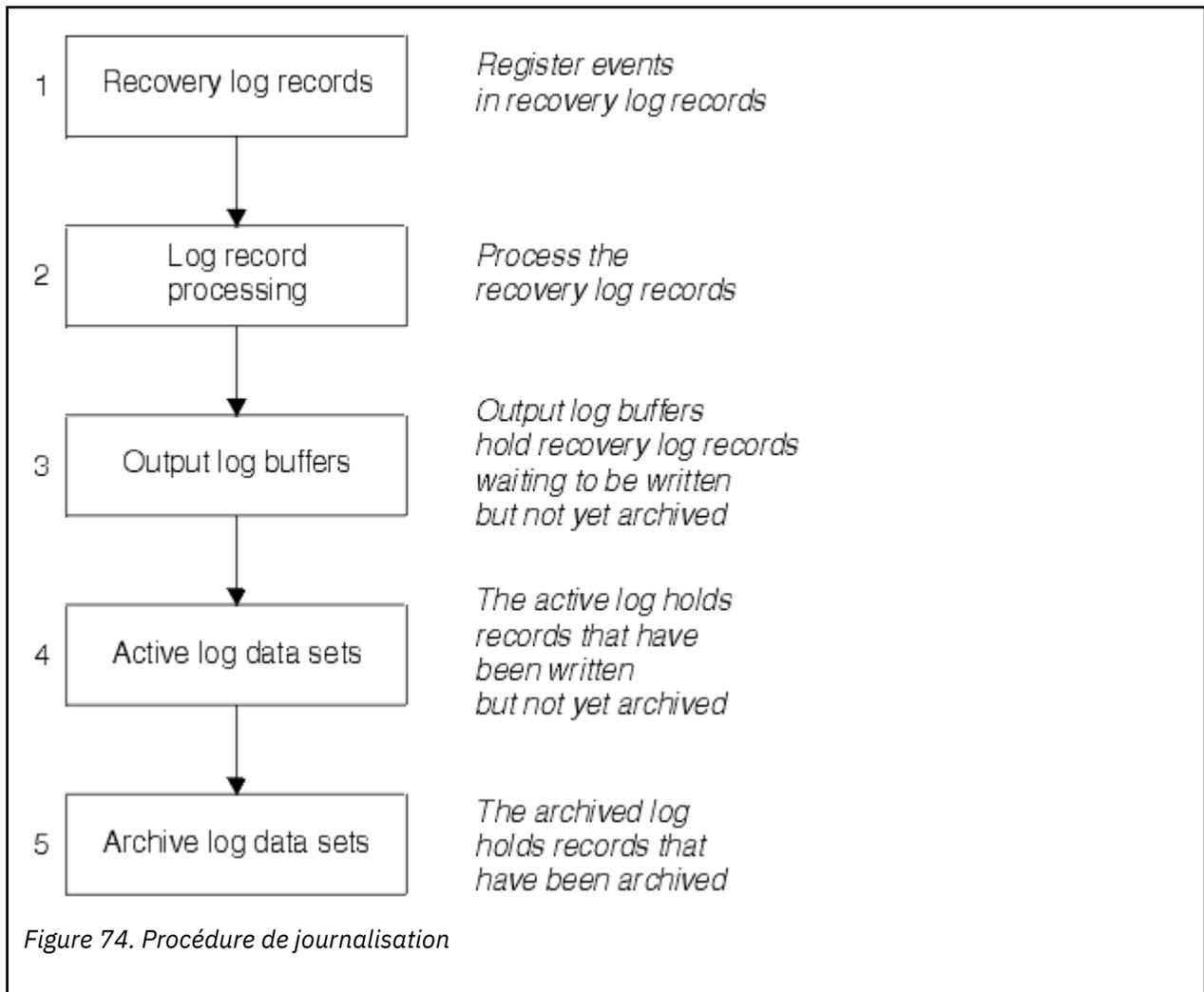
### Comment les journaux d'IBM MQ for z/OS sont-ils écrits ?

Consultez cette rubrique pour comprendre comment IBM MQ traite les enregistrements de fichier journal.

IBM MQ écrit chaque enregistrement journal dans un fichier de stockage à accès direct appelé *fichier actif*. Lorsque ce journal actif est saturé, IBM MQ copie son contenu vers un fichier de stockage à accès direct ou un jeu de données sur bande appelé *journal d'archivage*. Cette procédure s'appelle le *déchargement*.

La [Figure 74](#), à la [page 253](#) illustre la procédure de journalisation. Les enregistrements de journal suivent en général le cycle suivant :

1. IBM MQ consigne les modifications apportées aux données et les événements importants dans les enregistrements du journal de reprise.
2. IBM MQ traite les enregistrements du journal de reprise et les segmente, si nécessaire.
3. Les enregistrements de journal sont placés en séquence dans des *tampons du journal de sortie* formatés en tant qu'intervalles de contrôle VSAM. Chaque enregistrement de journal est identifié par une adresse relative en octets se trouvant dans l'intervalle compris entre zéro et  $2^{64}-1$ .
4. Les intervalles de contrôle sont consignés dans des jeux de données de journal actifs sur disque utilisés de manière séquentielle et recyclés.
5. Si l'archivage est actif, le contenu de chaque jeu de données de journal actif est automatiquement déchargé dans un nouveau fichier d'archivage au fur et à mesure qu'il se remplit.



## Écriture du journal actif

Un tampon du journal d'archivage est consigné dans le jeu de données de journal actif lorsque l'un des événements suivants se produit :

- Les tampons du journal sont saturés.
- Le seuil d'écriture est atteint (comme défini dans la macro CSQ6LOGP).
- Certains événements importants se produisent, par exemple, un point de validation ou l'émission d'une commande IBM MQ BACKUP CFSTRUCT.

Lorsque le gestionnaire de files d'attente est initialisé, les jeux de données de journal actifs nommés dans le fichier de stockage à accès direct sont alloués de manière dynamique pour utilisation exclusive par le gestionnaire de files d'attente et restent alloués à IBM MQ jusqu'à la fin de l'exécution du gestionnaire de files d'attente.

## Ajout dynamique de jeux de données de journal

Il est possible de définir de manière dynamique de nouveaux jeux de données de journal actifs lorsque le gestionnaire de files d'attente est en cours d'exécution. Cette fonction règle les éventuels problèmes liés à un blocage du gestionnaire de files d'attente lorsque la procédure d'archivage ne parvient pas à décharger les journaux actifs en raison d'un problème passager. Voir la commande [DEFINE LOG](#) pour plus d'informations.

**Remarque :** Pour redéfinir ou supprimer des journaux actifs, vous devez arrêter et redémarrer le gestionnaire de files d'attente.

## IBM MQ et sous-système de gestion de l'archivage (SMS)

Les paramètres IBM MQ permettent de définir les classes d'archivage Storage Management Subsystem (MVS/DFP SMS) lors de l'allocation dynamique des fichiers d'archivage IBM MQ. IBM MQ lance l'archivage des jeux de données de journal, mais vous pouvez utiliser SMS pour exécuter l'allocation du fichier d'archivage.

### Référence associée

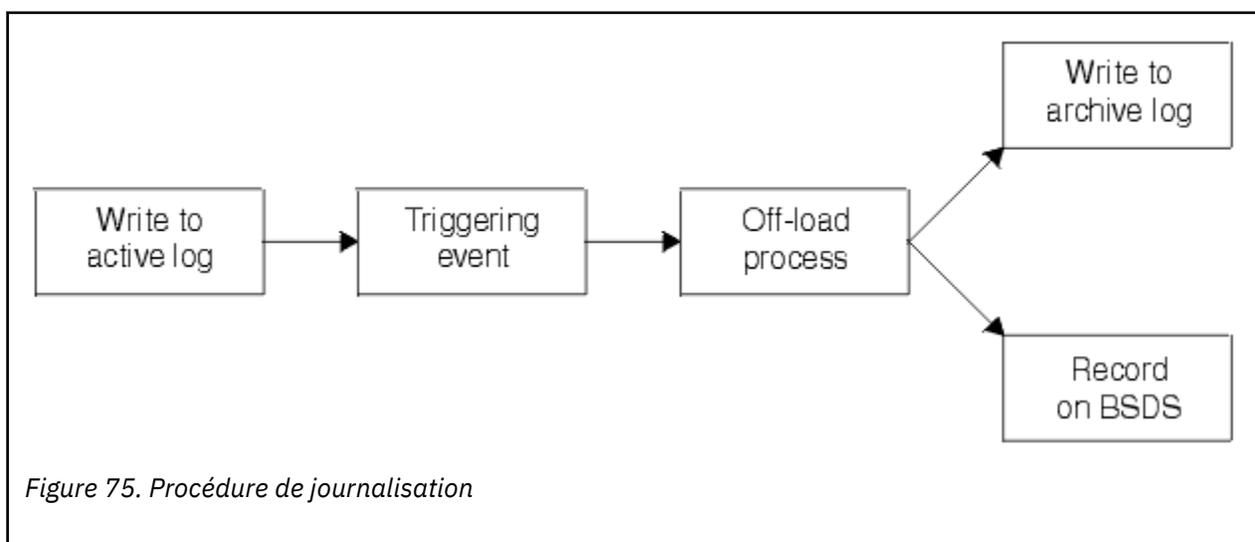
«A quel moment le journal d'archivage d'IBM MQ for z/OS est-il écrit ?», à la page 254

Consultez cette rubrique pour comprendre le processus de copie des journaux actif dans des journaux d'archivage et les cas dans lesquels ce processus intervient.

### **A quel moment le journal d'archivage d'IBM MQ for z/OS est-il écrit ?**

Consultez cette rubrique pour comprendre le processus de copie des journaux actif dans des journaux d'archivage et les cas dans lesquels ce processus intervient.

Le processus de copie des journaux actifs dans les journaux d'archive s'appelle le *déchargement*. La relation entre le déchargement et les autres événements de journalisation est présentée sous forme de schéma dans la Figure 75, à la page 254.



## Déclenchement du processus de déchargement

Le processus de déchargement d'un journal actif dans un journal d'archivage peut être déclenché par plusieurs événements. Exemple :

- Remplissage d'un jeu de données de journal actif.
- Utilisation de la commande MQSC ARCHIVE LOG.
- Erreur lors de l'écriture des données dans un jeu de données de journal actif.

Le jeu de données est tronqué avant le point de défaillance et le nouvel enregistrement non écrit devient le premier enregistrement du nouveau fichier. Le déchargement se déclenche pour le fichier tronqué de la même manière que pour un jeu de données de journal saturé normal. En cas de journaux actifs doubles, les deux copies sont tronquées de sorte à rester synchronisées.

Le message CSQJ110E est émis lorsque le dernier journal actif disponible est complet à 5% et à des incréments de 5% par la suite, indiquant le pourcentage de la capacité du journal en cours d'utilisation.

Si tous les journaux actifs sont saturés, IBM MQ arrête le traitement jusqu'au déchargement et émet ce message :

```
CSQJ111A +CSQ1 OUT OF SPACE IN ACTIVE LOG DATA SETS
```

## Processus de déchargement

Une fois que tous les journaux actifs sont saturés, IBM MQ exécute le processus de déchargement et arrête le traitement jusqu'à la fin de ce processus. Si le déchargement échoue lorsque les journaux actifs sont saturés, l'exécution d'IBM MQ se termine de façon anormale.

Lorsqu'un journal actif est prêt à être déchargé, une demande est envoyée à l'opérateur de la console z/OS pour monter une bande magnétique ou préparer un périphérique de stockage à accès direct. La valeur de l'option de journalisation ARCWTOR (pour plus d'informations, voir [Utilisation de CSQ6ARVP](#)) détermine si la demande peut être reçue. Si le déchargement se fait sur une bande, spécifiez `ARCWTOR=YES`. Si la valeur indiquée est YES, la demande est précédée par WTOR (numéro de message CSQJ008E), qui indique à l'opérateur de préparer un fichier d'archivage à allouer.

L'opérateur ne doit pas obligatoirement répondre immédiatement à ce message. Toutefois, une réponse tardive retarde le processus de déchargement. Les performances d'IBM MQ ne sont pas affectées, à moins que l'opérateur ne retarde sa réponse au point que les journaux actifs d'IBM MQ soient épuisés.

L'opérateur peut répondre en annulant le processus de déchargement. Dans ce cas, si l'allocation concerne la première copie de fichiers d'archivage doubles, le processus de déchargement est simplement retardé jusqu'à ce que le prochain jeu de données de journal actif soit saturé. Si l'allocation concerne la deuxième copie, le processus d'archivage bascule en mode unique, mais uniquement pour ce fichier.

## Interruptions et erreurs lors du déchargement

Aucune demande d'arrêt du gestionnaire de files d'attente ne prend effet tant que le déchargement n'est pas terminé. Si IBM MQ échoue lors du processus de déchargement, celui-ci reprend lors du redémarrage du gestionnaire de files d'attente.

## Messages émis lors du processus de déchargement

Les messages déchargés sont envoyés à la console z/OS par IBM MQ et le processus de déchargement. Vous pouvez utiliser ces messages pour rechercher les plages de RBA dans les différents fichiers journaux.

### Adresse relative en octets de journal de plus grande taille

Cette fonction augmente la disponibilité du gestionnaire de files d'attente en prolongeant la période avant laquelle une réinitialisation du journal est nécessaire.

Les données de reprise sont écrites dans le journal afin que les messages persistants soient disponibles lors du redémarrage du gestionnaire de files d'attente. Le terme adresse relative en octets de journal est utilisé pour faire référence à l'emplacement des données en tant que décalage par rapport au début du journal.

Avant IBM MQ 8.0, l'adresse relative en octets de journal de 6 octets pouvait adresser jusqu'à 256 téraoctets de données. Pour que cette quantité d'enregistrements de journal puisse être écrite, vous devez réinitialiser le journal du gestionnaire de files d'attente en suivant la procédure décrite dans [Réinitialisation du journal du gestionnaire de files d'attente](#).

La réinitialisation des journaux des gestionnaires de files d'attente n'est pas un processus rapide et peut entraîner une longue indisponibilité car les ensembles de pages doivent être réinitialisés au cours de l'opération. Pour un gestionnaire de files d'attente très utilisé, cette opération peut être effectuée généralement une fois par an.

Depuis IBM MQ 8.0, la longueur de l'adresse relative en octets de journal peut être de 8 octets et le gestionnaire de files d'attente peut maintenant adresser 64 000 fois plus de données (16 exaoctets) avant qu'il soit nécessaire de réinitialiser l'adresse relative en octets de journal. L'utilisation d'une adresse relative en octets de journal plus longue a pour conséquence une augmentation de quelques octets de la taille des données de journal écrites.

## Quand cette fonction est-elle activée ?

**LTS** Vous pouvez activer cette fonction à tout moment, bien qu'il soit préférable de planifier la modification et de l'activer explicitement au moment opportun pour l'entreprise, avant que l'adresse relative en octets de journal en cours n'atteigne la plage limite de l'adresse relative en octets de journal de 6 octets. Voir [Planification de l'augmentation de la plage maximales de journaux adressables](#) pour plus d'informations sur la planification de la migration vers des adresses relatives en octets de journal de 8 octets.

**V 9.2.5** Les gestionnaires de files d'attente créés pour IBM MQ 9.2.5 ou une version ultérieure ont déjà activé cette fonction.

**Important :** pour que vous puissiez utiliser cette fonction dans un groupe de partage de files d'attente, tous les gestionnaires de files d'attente dans le groupe de partage de files d'attente doivent être à l'un des niveaux suivants :

- A la IBM MQ 9.0.n CD, IBM MQ 9.1.0 LTS, ou à une version ultérieure
- A la IBM MQ 9.0.0 et démarrés avec **OPMODE=(NEWFUNC,800)** ou **OPMODE=(NEWFUNC,900)**

Si la plage limite est presque atteinte par l'adresse relative en octets de journal en cours, envisagez de convertir le gestionnaire de files d'attente de sorte qu'il utilise une adresse relative en octets de journal de 8 octets au lieu de réinitialiser le journal du gestionnaire de files d'attente. Cette opération de conversion entraîne une indisponibilité plus courte que la réinitialisation du journal et prolonge significativement la période avant laquelle cette dernière doit être effectuée.

Le message CSQJ034I, émis au cours de l'initialisation du gestionnaire de files d'attente, indique la fin de la plage d'adresses relatives en octets du journal du gestionnaire de files d'attente, conformément à la configuration, et permet de déterminer si des adresses relatives en octets de 6 octets ou de 8 octets du journal sont utilisées.

## Comment cette fonction est-elle activée ?

L'adresse relative en octets de journal de 8 octets est activée en démarrant le gestionnaire de files d'attente avec un fichier d'amorce au format version 2. En bref, la procédure est la suivante :

1. Vérification du respect par tous les gestionnaires de files d'attente du groupe de partage de files d'attente des exigences pour l'activation d'une adresse relative en octets de journal de 8 octets
2. Arrêt du gestionnaire de files d'attente
3. Exécution de l'[utilitaire de conversion du fichier d'amorce](#) pour créer une copie de ce fichier au format version 2
4. Redémarrage du gestionnaire de files d'attente avec le fichier d'amorce converti

Une fois converti pour utiliser des adresses relatives en octets de journal de 8 octets, un gestionnaire de files d'attente ne peut plus utiliser une adresse relative en octets de journal de 6 octets.

Voir [Implémentation de la plus grande adresse relative en octets de journal](#) pour plus d'informations sur l'activation des adresses relatives en octets de journal de 8 octets.

### Tâches associées

[Planification de l'augmentation de la plage de données de journal adressable maximale](#)

### Référence associée

[Utilitaire de conversion du fichier d'amorce \(CSQJUCNV\)](#)

Le fichier d'amorce est nécessaire à IBM MQ comme mécanisme pour référencer les fichiers journaux et les enregistrements de journal. Les informations qu'il contient sont nécessaires lors du traitement normal des opérations et en cas de reprise par redémarrage.

### Finalité du fichier d'amorce

Le *fichier d'amorce* et un fichier VSAM KSDS (key-sequenced data set) qui contient les informations nécessaires à IBM MQ. Il contient les éléments suivants :

- un inventaire de tous les fichiers actifs et archivés connus d'IBM MQ. IBM MQ utilise cet outil d'inventaire pour :
  - Assurer le suivi des fichiers journaux actifs et archivés
  - Localiser les enregistrements de journal afin de pouvoir satisfaire les demandes de lecture des journaux en cours de traitement normal
  - Localiser les enregistrements de journal afin de gérer le traitement des redémarrages

IBM MQ stocke les informations dans l'inventaire à chaque fois qu'un jeu de données de journal d'archivage est défini ou qu'un jeu de données de journal actif est réutilisé. Pour les journaux actifs, l'inventaire présente les journaux déjà complets et les journaux pouvant être réutilisés. Il contient l'adresse relative en octets de chaque partie du journal contenue dans ce fichier.

- un inventaire *de bouclage* de toutes les activités IBM MQ récentes. Cet inventaire est nécessaire si vous devez redémarrer le gestionnaire de files d'attente.

Le fichier d'amorce est nécessaire si une erreur liée au gestionnaire de files d'attente se produit et si vous devez redémarrer celui-ci. IBM MQ **doit** avoir un fichier d'amorce. Pour minimiser les risques de problème lors d'un redémarrage, vous pouvez configurer IBM MQ avec deux fichiers d'amorce destinés à enregistrer les mêmes informations. L'utilisation d'un fichier d'amorce double est appelée exécution en *mode dual*. Placez si possible les deux copies sur des volumes distincts. Vous réduirez ainsi le risque de perdre ces deux copies au cas où le volume serait détruit ou endommagé. Utilisez un fichier d'amorce double plutôt qu'une écriture en double sur un disque.

Le fichier d'amorce est configuré lorsqu'IBM MQ est personnalisé, et vous pouvez gérer l'inventaire à l'aide de l'utilitaire de modification de l'inventaire du journal (CSQJU003). Pour plus d'informations sur cet utilitaire, voir [Administration de IBM MQ for z/OS](#). Il est référencé par une instruction de définition de données dans la procédure de démarrage du gestionnaire de files d'attente.

Normalement, IBM MQ conserve deux copies du fichier d'amorce. Si une erreur d'E-S se produit, l'allocation de la copie défaillante est annulée et les opérations se poursuivent avec un seul fichier d'amorce. Vous pouvez restaurer le mode dual. Voir la description de cette restauration à la section [Administration de IBM MQ for z/OS](#).

Les journaux actifs sont d'abord enregistrés dans le fichier d'amorce lors de l'installation d'IBM MQ. Vous ne pouvez pas remplacer les journaux actifs sans arrêter, puis redémarrer le gestionnaire de files d'attente.

Les fichiers d'archivage sont alloués de manière dynamique. Lorsqu'un journal est alloué, son nom est enregistré dans le fichier d'amorce. La liste des fichiers d'archivage s'allonge au fur et à mesure que des archives sont ajoutées. Lorsqu'un nombre défini par l'utilisateur d'entrées a été entré, cette liste entre dans une boucle. Le nombre maximal d'entrées est égal à 1 000 pour la journalisation simple et 2 000 pour la journalisation double.

Vous pouvez utiliser un système de gestion de bandes pour supprimer les fichiers d'archivage (IBM MQ ne propose aucune méthode automatisée). Les informations relatives à un fichier d'archivage peuvent rester dans le fichier d'amorce longtemps après la suppression de ce fichier par l'administrateur système.

Inversement, le nombre maximal de fichiers d'archivage peut avoir été dépassé et les données du fichier d'amorce peuvent avoir été supprimées bien avant la date d'expiration du fichier.

Vous pouvez utiliser la commande MQSC pour définir l'étendue du journal et le nom du fichier d'archivage ou du fichier journal actif contenant l'adresse relative en octets la plus ancienne. Ces éléments sont requis pour différents types de récupération de support ou de gestionnaire de files d'attente.

```
DISPLAY USAGE TYPE(DATASET)
```

Si le module des paramètres système définit que les fichiers d'archivage sont catalogués lors de leur allocation, le fichier d'amorce pointe vers le catalogue de la fonction de catalogage intégrée pour obtenir les informations nécessaires aux allocations suivantes. Sinon, les entrées du fichier d'amorce de chaque volume enregistrent le numéro de série du volume et les informations sur l'unité nécessaires aux allocations suivantes.

## Version du fichier d'amorce

Le format du fichier d'amorce varie selon la version. L'augmentation de la version du fichier d'amorce permet d'utiliser de nouvelles fonctions. Les versions suivantes de fichier d'amorce sont prises en charge par IBM MQ :

### Version 1

Prise en charge par toutes les éditions d'IBM MQ. Un fichier d'amorce version 1 prend en charge des valeurs d'adresse relative en octets de 6 octets.

### Version 2

Prise en charge par IBM MQ 8.0 et les versions ultérieures. Un fichier d'amorce version 2 accepte des valeurs d'adresse relative en octets de 8 octets et jusqu'à 310 fichiers dans chaque copie de fichier journal actif.

**V 9.2.5** Activé par défaut pour les gestionnaires de files d'attente créés dans IBM MQ 9.2.4 et dans les versions ultérieures.

### version 3

Prise en charge par IBM MQ 8.0 et les versions ultérieures. Le fichier BSDS est automatiquement converti en version 3, à partir de la version 2, lorsque plus de 31 ensembles de données sont ajoutés à l'une des copies du journal actif.

Vous pouvez déterminer la version du fichier d'amorce en exécutant l'utilitaire Print Log Map ([CSQJU004](#)). Pour convertir un fichier d'amorce de la version 1 à la version 2, exécutez l'utilitaire de conversion correspondant ([CSQJUCNV](#)).

Pour plus d'informations sur les fichiers d'amorce de journal de 6 et 8 octets, voir [«Adresse relative en octets de journal de plus grande taille»](#), à la page 255.

## Fichiers d'archivage et copies du fichier d'amorce

A chaque fois qu'un fichier d'archivage est créé, une copie du fichier d'amorce est également créée. Si le journal d'archivage se trouve sur une bande, le fichier d'amorce est le premier fichier du premier volume de sortie. S'il se trouve sur un disque, le fichier d'amorce est un fichier distinct.

Les noms du fichier d'archivage et de la copie du fichier d'amorce sont les mêmes, à cette exception que le qualificatif de niveau inférieur du journal d'archivage commence par A que le nom de la copie du fichier d'amorce commence par B, par exemple :

### Nom du journal d'archivage

```
CSQ.ARCHLOG1.E00186.T2336229. A 0000001
```

### Nom de la copie du fichier d'amorce

```
CSQ.ARCHLOG1.E00186.T2336229. B 0000001
```

En cas d'erreur de lecture lors de la copie du fichier d'amorce, la copie n'est pas créée, le message [CSQJ125E](#) est émis et le déchargement du nouveau fichier d'archivage se poursuit sans la copie du fichier d'amorce.

## Définition de votre système sur z/OS

IBM MQ for z/OS utilise plusieurs définitions d'objet par défaut et fournit des exemples de JCL permettant de créer ces objets. Cette rubrique permet de comprendre les objets par défaut et l'exemple de JCL.

### Définition des paramètres système

Dans IBM MQ for z/OS, un module de paramètres système contrôle les environnements de consignation, d'archivage, de traçage et de connexion utilisés par IBM MQ. Ces paramètres système sont définis par les trois macros assembleur suivantes :

#### CSQ6SYSP

Paramètres système, notamment définition de l'environnement de connexion et de traçage.

#### CSQ6LOGP

Paramètres de journalisation.

#### CSQ6ARVP

Paramètres d'archivage des journaux.

Des modules de paramètres par défaut sont fournis avec IBM MQ for z/OS. S'ils ne contiennent pas les valeurs que vous souhaitez utiliser, vous pouvez créer vos propres modules de paramètres à l'aide de l'exemple fourni avec IBM MQ. L'exemple est `th1qua1.SCSQPROC (CSQ4ZPRM)`.

Vous pouvez modifier certains paramètres système lorsqu'un gestionnaire de files d'attente est en cours d'exécution. Voir les commandes SET SYSTEM, SET LOG et SET ARCHIVE dans [Les commandes MQSC](#).

Pour plus d'informations sur la définition, consultez les rubriques suivantes :

- [«Définition des objets système pour IBM MQ for z/OS», à la page 259](#)
- [«Optimisation de votre gestionnaire de files d'attente sous IBM MQ for z/OS», à la page 264](#)
- [«Exemples de définition fournis avec IBM MQ for z/OS», à la page 265](#)

### Concepts associés

[Sources à partir desquelles vous pouvez émettre des commandes MQSC sous z/OS](#)

### Tâches associées

[Personnalisation des exemples de fichiers en entrée d'initialisation](#)

[Administration de z/OS](#)

[Configuration des clusters](#)

[Surveillance d'IBM MQ](#)

## Définition des objets système pour IBM MQ for z/OS

IBM MQ for z/OS requiert des objets prédéfinis supplémentaires pour les applications de publication/abonnement, pour le contrôle des clusters et des canaux et pour d'autres fonctions d'administration du système.

Les objets système requis par IBM MQ for z/OS peuvent être divisés en différentes catégories :

- [Objets de publication/abonnement](#)
- [Objets par défaut du système](#)
- [Objets de commande du système](#)
- [Objets d'administration du système](#)
- [Files d'attente de canaux](#)
- [Files d'attente de cluster](#)
- [Files d'attente de groupe de partage de files d'attente](#)
- [Classes d'archivage](#)
- [Définition de la file d'attente de rebut des objets système](#)

- [File d'attente de transmission par défaut](#)
- [Files d'attente internes](#)
- [«File d'attente des authentifications de canal», à la page 264](#)

## Objets de publication/abonnement

Vous devez définir plusieurs objets système avant de pouvoir utiliser les applications de publication/abonnement avec IBM MQ for z/OS. Des exemples de définition destinés à vous aider dans cette tâche sont fournis avec IBM MQ. Ces exemples sont décrits à la section [CSQ4INSG](#).

Pour utiliser la fonction de publication/abonnement, vous devez définir les objets suivants :

- Une file d'attente locale nommée SYSTEM.RETAINED.PUB.QUEUE utilisée pour contenir une copie de chaque publication conservée dans le gestionnaire de files d'attente. Chaque nom complet de rubrique peut contenir une publication conservée stockée dans cette file d'attente au maximum. Si vos applications utilisent les publications conservées dans plusieurs rubriques différentes ou si les messages de ces publications sont volumineux, vous devez planifier soigneusement les besoins de cette file d'attente en matière de stockage et envisager de l'affecter à son propre ensemble de pages si ces besoins sont trop importants. Pour améliorer les performances, vous devez définir cette file d'attente avec le type d'index MSGID (comme dans l'exemple de définition de file d'attente fourni).
- Une file d'attente locale nommée SYSTEM.DURABLE.SUBSCRIBER.QUEUE utilisée pour contenir une copie persistante des abonnements durables du gestionnaire de files d'attente. Pour améliorer les performances, vous devez définir cette file d'attente avec le type d'index CORRELID (comme dans l'exemple de définition de file d'attente fourni).
- Une file d'attente locale nommée SYSTEM.DURABLE.MODEL.QUEUE utilisée en tant que modèle d'abonnement durable géré.
- Une file d'attente locale nommée SYSTEM.NDURABLE.MODEL.QUEUE utilisée en tant que modèle d'abonnement non durable géré.
- Une liste de noms nommée SYSTEM.QPUBSUB.QUEUE.NAMELIST contenant une liste des noms de file d'attente contrôlée par l'interface de publication/abonnement mise en file d'attente.
- Une liste de noms nommée SYSTEM.QPUBSUB.SUBPOINT.NAMELIST contenant une liste des objets de rubrique utilisée par l'interface de publication/abonnement mis en file d'attente pour établir une correspondance entre les objets de rubrique et les points d'abonnement.
- Une rubrique nommée SYSTEM.BASE.TOPIC utilisée en tant que rubrique de base pour la résolution des attributs.
- Une rubrique nommée SYSTEM.BROKER.DEFAULT.STREAM constituant le flot par défaut utilisé par l'interface de publication/abonnement mise en file d'attente.
- Une rubrique nommée SYSTEM.BROKER.DEFAULT.SUBPOINT constituant le point d'abonnement RFH2 par défaut utilisé par l'interface de publication/abonnement mise en file d'attente.
- Une rubrique nommée SYSTEM.BROKER.ADMIN.STREAM constituant le flot administratif utilisé par l'interface de publication/abonnement mise en file d'attente.
- Un abonnement nommé SYSTEM.DEFAULT.SUB constituant un objet d'abonnement par défaut utilisé pour fournir les valeurs par défaut aux commandes DEFINE SUB.

## Objets par défaut du système

Les objets par défaut du système permettent de fournir des attributs par défaut lorsque vous définissez un objet sans utiliser un autre nom d'objet comme référence.

Les noms des définitions d'objet système par défaut commencent par les caractères "SYSTEM.DEFAULT" ou "SYSTEM.DEF." La file d'attente locale par défaut du système se nomme par exemple SYSTEM.DEFAULT.LOCAL.QUEUE.

Ces objets définissent les valeurs par défaut du système pour les attributs des objets IBM MQ MQ suivants :

- Files d'attente locales
- Files d'attente modèle
- Files d'attente alias
- Files d'attente éloignées
- Processus
- Listes de noms
- Canaux
- Classes d'archivage
- Informations d'authentification

Les files d'attente partagées constituent un type spécial de file d'attente local. Elles sont définies à partir de `SYSTEM.DEFAULT.LOCAL.QUEUE`. N'oubliez pas d'entrer une valeur pour le nom de l'unité de couplage, car ce nom ne figure pas dans la définition par défaut. Vous pouvez également définir votre propre définition de file d'attente partagée par défaut et l'utiliser comme base pour les files d'attente partagées. Toutes les files d'attente de ce type hériteront donc des attributs requis. N'oubliez pas que vous devez définir une file d'attente partagée sur un seul gestionnaire de files d'attente du groupe de partage de files d'attente.

## Objets de commande du système

Les noms des objets de commande du système commencent par les caractères `SYSTEM.COMMAND`. Vous devez définir ces objets pour pouvoir utiliser les panneaux d'opérations et de contrôle IBM MQ pour émettre des commandes vers un sous-système IBM MQ.

Il existe deux objets de commande du système :

1. La file d'attente d'entrée des commandes système est une file d'attente locale dans laquelle les commandes sont placées avant d'être traitées par le processeur de commandes IBM MQ. Elle doit être nommée `SYSTEM.COMMAND.INPUT`. Un alias nommé `SYSTEM.ADMIN.COMMAND.QUEUE` doit également être défini, pour la compatibilité avec IBM MQ for Multiplatformset pour une utilisation par MQ Console et administrative REST API.
2. `SYSTEM.COMMAND.REPLY.MODEL` est une file d'attente modèle qui définit la file d'attente de réponse des commandes du système.

Deux objets supplémentaires peuvent également être utilisés par IBM MQ Explorer :

- La file d'attente `SYSTEM.MQEXPLORER.REPLY.MODEL`
- Le canal `SYSTEM.ADMIN.SVRCONN`

`SYSTEME SYSTEM.REST.REPLY.QUEUE` est la file d'attente de réponses utilisée par IBM MQ administrative REST API.

Les commandes sont normalement envoyées à l'aide de messages non persistants. Les objets de commande du système doivent donc être associés à l'attribut `DEFPSIST(NO)` de sorte que les applications qui les utilisent (notamment les applications fournies telles que le programme utilitaire et les panneaux d'opérations et de commandes) reçoivent les messages non persistants par défaut. Si l'une des applications utilise des messages persistants pour les commandes, définissez l'attribut `DEFTYPE(PERMDYN)` pour la file d'attente de réponse, car les messages de réponse à ces commandes sont persistants.

## Objets d'administration du système

Les noms des objets d'administration du système commencent par les caractères `SYSTEM.ADMIN`.

Il existe sept objets d'administration du système :

- La file d'attente SYSTEM.ADMIN.CHANNEL.EVENT
- La file d'attente SYSTEM.ADMIN.COMMAND.EVENT
- La file d'attente SYSTEM.ADMIN.CONFIG.EVENT
- La file d'attente SYSTEM.ADMIN.PERFM.EVENT
- La file d'attente SYSTEM.ADMIN.QMGR.EVENT
- La file d'attente SYSTEM.ADMIN.TRACE.ROUTE.QUEUE
- La file d'attente SYSTEM.ADMIN.ACTIVITY.QUEUE

## Files d'attente de canaux

Pour utiliser les files d'attente réparties, vous devez définir les objets suivants :

- Une file d'attente locale nommée SYSTEM.CHANNEL.SYNCQ utilisée pour gérer les numéros de séquence et les identificateurs d'unités d'oeuvre logiques des canaux. Pour améliorer les performances du canal, vous devez définir cette file d'attente avec le type d'index MSGID (comme dans l'exemple de définition de file d'attente fourni).
- Une file d'attente locale nommée SYSTEM.CHANNEL.INITQ utilisée pour les commandes relatives aux canaux.

Il est impossible de définir ces files d'attente comme des files d'attente partagées.

## Files d'attente de cluster

Pour utiliser les clusters IBM MQ, vous devez définir les objets suivants :

- Une file d'attente locale nommée SYSTEM.CLUSTER.COMMAND.QUEUE utilisée pour communiquer les modifications apportées au référentiel entre les gestionnaires de files d'attente. Les messages écrits dans cette file d'attente contiennent les mises à jour des données du référentiel devant être appliquées à la copie locale du référentiel ou les demandes de données du référentiel.
- Une file d'attente locale nommée SYSTEM.CLUSTER.REPOSITORY.QUEUE utilisée pour conserver une copie persistante du référentiel.
- Une file d'attente locale nommée SYSTEM.CLUSTER.TRANSMIT.QUEUE constituant la file d'attente de transmission pour toutes les destinations du cluster. Pour des raisons liées aux performances, vous devez définir cette file d'attente avec le type d'index CORRELID (comme dans l'exemple de définition de file d'attente fourni).

Ces files d'attente contiennent en général un grand nombre de messages.

Il est impossible de définir ces files d'attente comme des files d'attente partagées.

## Files d'attente de groupe de partage de files d'attente

Pour utiliser les canaux partagés et les files d'attente intra-groupes, vous devez définir les objets suivants :

- Une file d'attente partagée nommée SYSTEM.QSG.CHANNEL.SYNCQ utilisée pour contenir les informations de synchronisation pour les canaux partagés.
- Une file d'attente partagée nommée SYSTEM.QSG.TRANSMIT.QUEUE utilisée en tant que file d'attente de transmission pour les files d'attente intra-groupe. Si vos opérations s'exécutent dans un groupe de partage de files d'attente, vous devez définir cette file d'attente, même si vous n'utilisez pas les files d'attente intra-groupe.

## Classes d'archivage

Nous vous recommandons de définir les six classes d'archivage suivantes. Quatre d'entre elles sont requises par IBM MQ. Nous vous recommandons de créer les deux autres définitions car elles sont utilisées dans les exemples de définition de file d'attente.

### **DEFAULT (obligatoire)**

Cette classe d'archivage est utilisée par toutes les files d'attente de messages qui ne sont pas d'une importance critique pour les performances et qui ne correspondent à aucune autre classe d'archivage. Elle constitue la classe d'archivage par défaut si vous n'en indiquez aucune lorsque vous définissez une file d'attente.

### **NODEFINE (obligatoire)**

Cette classe d'archivage est utilisée si aucune classe d'archivage n'est indiquée lorsque vous définissez une file d'attente.

### **REMOTE (obligatoire)**

Cette classe d'archivage est principalement utilisée pour les files d'attente de transmission, c'est-à-dire les files d'attente associées au système contenant des messages de courte durée et d'une importance critique pour les performances.

### **SYSNLGLV**

Cette classe d'archivage est utilisée pour les messages de longue durée et d'une importance critique pour les performances.

### **SYSTEM (obligatoire)**

Cette classe de stockage est utilisée pour les files d'attente de messages système critiques pour les performances, par exemple SYSTEM.CHANNEL.SYNQ et SYSTEM.CLUSTER.\* files d'attente.

### **SYSVOLAT**

Cette classe d'archivage est utilisée pour les messages de courte durée et d'une importance critique pour les performances.

Vous pouvez si nécessaire modifier leurs attributs et ajouter d'autres définitions de classe d'archivage.

## Définition de la file d'attente de rebut des objets système

La file d'attente de rebut est utilisée si la destination du message n'est pas valide. IBM MQ place ce message dans une file d'attente nommée file d'attente de rebut. Bien qu'elle ne soit pas obligatoire, vous devez la considérer comme essentielle, notamment si vous utilisez les files d'attente réparties ou l'une des passerelles IBM MQ.

Ne définissez **pas** la file d'attente de rebut comme une file d'attente partagée. L'insertion dans une file d'attente locale d'un gestionnaire de files d'attente peut aboutir dans la file d'attente de rebut. Si la file d'attente de rebut était une file d'attente partagée, un gestionnaire de files d'attente de rebus sur un système différent peut traiter le message et le placer dans une file d'attente du même nom, mais comme celle-ci appartient à un gestionnaire de files d'attente différent, il ne s'agit pas de la bonne file d'attente ou le profil de sécurité est différent. Si la file d'attente n'existait pas, le retraitement échouerait.

Si vous décidez de définir une file d'attente de rebut, vous devez également communiquer son nom au gestionnaire de files d'attente. Pour cela, utilisez la commande ALTER QMGR DEADQ(*nom\_file\_attente*). Pour plus d'informations, voir [Displaying and altering queue manager attributes](#).

## File d'attente de transmission par défaut

La file d'attente de transmission par défaut est la file d'attente utilisée si aucune autre file d'attente de transmission n'est disponible pour envoyer des messages à un autre gestionnaire de files d'attente. Si vous définissez une file d'attente de transmission par défaut, vous devez également définir un canal pour la servir. A défaut, les messages placés dans la file d'attente de transmission par défaut ne sont pas transmis au gestionnaire de files d'attente éloignées et restent dans la file d'attente.

Si vous décidez de définir une file d'attente de transmission par défaut, vous devez également communiquer son nom au gestionnaire de files d'attente. Pour cela, utilisez la commande ALTER QMGR.

## Files d'attente internes

### • File d'attente des données en attente

- Une file d'attente définie pour une utilisation interne, SYSTEM.PENDING.DATA.QUEUE, prend en charge l'utilisation d'abonnements durables dans un environnement de publication / abonnement JMS .

### • File d'attente de transfert de retard de distribution JMS 2.0

- Si la fonctionnalité de délai de distribution fournie par JMS 2.0 est utilisée, une file d'attente de transfert interne, SYSTEM.DDELAY.LOCAL.QUEUE, doit être définie. Cette file d'attente est utilisée par le gestionnaire de files d'attente pour stocker temporairement les messages, envoyés avec un délai de distribution différent de zéro, jusqu'à expiration du délai de distribution. Les messages sont ensuite envoyés à leur destination cible. Vous trouverez un exemple de définition de ce type de file d'attente dans CSQ4INSG, en commentaire.
- Lorsque vous définissez la file d'attente SYSTEM.DDELAY.LOCAL.QUEUE, vous devez définir les attributs STGCLASS, MAXMSGL et MAXDEPTH pour le nombre anticipé de messages qui seront envoyés avec un délai de distribution. De plus, lorsque vous définissez la file d'attente SYSTEM.DDELAY.LOCAL.QUEUE, assurez-vous que le gestionnaire de files d'attente peut placer les messages dans cette file d'attente. Vous devez veiller à ce qu'aucun ID utilisateur n'ait les droits permettant de placer des messages dans cette file d'attente.

## File d'attente des authentifications de canal

La file d'attente SYSTEM.CHLAUTH.DATA.QUEUE est nécessaire en cas d'usage interne de l'authentification de canal. Des exemples de définition destinés à vous aider dans cette tâche sont fournis avec IBM MQ. Ces exemples sont décrits dans CSQ4INSA, qui définit également certaines règles par défaut.

## Optimisation de votre gestionnaire de files d'attente sous IBM MQ for z/OS

Vous pouvez exécuter quelques étapes simples afin d'optimiser votre gestionnaire de files d'attente et d'éviter certains problèmes de base liés aux performances.

Il existe différentes manières d'améliorer les performances de votre gestionnaire de files d'attente, qui sont contrôlées par les attributs du gestionnaire de files d'attente défini par la commande ALTER QMGR. Cette section contient des informations sur les différentes méthodes permettant de procéder à cette amélioration en définissant le nombre maximal de messages autorisés sur le gestionnaire de files d'attente ou en exécutant des tâches de nettoyage sur celui-ci. [IBM MQ SupportPac MP16 - WebSphere MQ for z/OS Planification et optimisation de la capacité](#) fournit plus d'informations sur les performances et l'optimisation.

## Points de synchronisation

Le contrôle des points de synchronisation dans une application est l'un des rôles du gestionnaire de files d'attente. L'application construit une unité d'oeuvre contenant n'importe quel nombre d'appels MQPUT ou MQGET se terminant par un appel MQCMIT.

Comme le nombre d'appels MQPUT ou MQGET dans la portée d'un MQCMIT augmente, le coût de performance de la validation augmente de manière significative. Les applications, en général, doivent être

conçues pour ne pas entraîner de MQPUT/MQGET pour un grand nombre de messages dans un point de synchronisation unique.

Vous pouvez limiter administrativement le nombre de messages dans un point de synchronisation unique à l'aide de l'attribut MAXUMSGS du gestionnaire de files d'attente. Si une application dépasse cette limite, elle reçoit le message MQRC\_SYNCPOINT\_LIMIT\_REACHED sur l'appel MQPUT, MQPUT1 ou MQGET qui dépasse la limite. L'application doit alors émettre MQCMIT ou MQBACK selon le cas.

La valeur par défaut de MAXUMSGS est 10000. Cette valeur peut être abaissée si vous souhaitez appliquer une limite inférieure, ce qui peut également aider à protéger contre les applications tournant en boucle. Avant de réduire MAXUMSGS, assurez-vous de bien comprendre vos applications existantes pour vous assurer qu'elles ne dépassent pas la limite ou qu'elles peuvent accepter le code retour MQRC\_SYNCPOINT\_LIMIT\_REACHED

## Messages arrivés à expiration

Les messages ayant expiré sont supprimés par l'appel MQGET approprié suivant. Si aucun appel de ce type ne se produit, les messages expirés ne sont pas supprimés et, pour certaines files d'attente, notamment celles où l'extraction des messages se fait par l'ID message, l'ID corrélation ou l'ID groupe et où la file d'attente est indexée en vue des performances, plusieurs messages arrivés à expiration peuvent se cumuler. Le gestionnaire de files d'attente peut rechercher régulièrement les messages arrivés à expiration dans les files d'attente, puis les supprimer. Vous pouvez choisir la fréquence à laquelle cette recherche se fait. Pour cela, procédez de l'une des deux manières suivantes :

### Demande explicite

Vous pouvez contrôler les files d'attente dans lesquelles vous exécutez des recherches et à quel moment vous les exécutez. Emettez la commande REFRESH QMGR TYPE(EXPIRY) en spécifiant la ou les files d'attente dans lesquelles vous souhaitez exécuter la recherche.

### Recherche périodique

Vous pouvez spécifier un intervalle d'expiration dans l'objet gestionnaire de files d'attente à l'aide de l'attribut EXPRYINT. Le gestionnaire de files d'attente gère les informations relatives aux messages arrivés à expiration sur chaque file d'attente et sait à quel moment il est recommandé de rechercher les messages arrivés à expiration. A chaque fois que l'intervalle EXPRYINT est atteint, le gestionnaire de files d'attente recherche les files d'attente dans lesquelles il est recommandé de rechercher les messages arrivés à expiration et lance la recherche dans celles-ci, et non dans toutes les files d'attente. Ainsi, toute perte de temps processeur liée à des recherches inutiles est évitée.

Un seul gestionnaire de files d'attente du groupe de partage de files d'attente exécute la recherche dans les files d'attente. En général, le premier gestionnaire de files d'attente qui redémarre ou pour lequel l'intervalle EXPRYINT est défini exécute la recherche.

**Remarque :** vous devez définir la même valeur pour EXPRYINT pour tous les gestionnaires de files d'attente d'un même groupe de partage de files d'attente.

## Exemples de définition fournis avec IBM MQ for z/OS

Utilisez cette rubrique comme référence pour l'exemple de JCL et le code fourni avec IBM MQ for z/OS.

Les exemples de définition suivants sont fournis avec IBM MQ dans la bibliothèque thlqual.SCSQPROC. Vous pouvez les utiliser pour définir les objets système et pour personnaliser vos propres objets. Vous pouvez en inclure certains dans les fichiers d'initialisation en entrée (décrits à la section [Commandes d'initialisation](#)).

Fichier d'initialisation en entrée	Nom de l'exemple
<a href="#">CSQINP1</a>	CSQ4INP1 CSQ4INPR

Tableau 21. Exemples de définition IBM MQ pour les objets système (suite)

Fichier d'initialisation en entrée	Nom de l'exemple
<u>CSQINP2</u>	CSQ4INSA CSQ4INYS <sup>1</sup> CSQ4INSX CSQ4INSG CSQ4INSR CSQ4INSS CSQ4INSJ CSQ4INSM CSQ4INYG CSQ4INYR CSQ4INYC CSQ4INYD CSQ4INSC
<u>CSQINPT</u>	CSQ4INST CSQ4INYT
<u>Autre</u>	CSQ4DISP CSQ4INPX CSQ4IVPQ CSQ4IVPG CSQ4MSTR CSQ4MSRR V 9.2.0 CSQ4QMIN

**Remarque :**

1. L'ordre de ces exemples de définition est important : une erreur se produit si INYS, INSX et INSG ne sont pas ordonnés correctement.

**Exemples CSQINP1**

Utilisez l'exemple de fichier CSQINP1 thlqual.SCSQPROC(CSQ4INP1) lorsque vous utilisez un ensemble de pages pour chaque classe de message ou thlqual.SCSQPROC(CSQ4INPR) lorsque vous utilisez plusieurs ensembles de pages pour les principales classes de message. Cet exemple de fichier contient des définitions des pools de mémoire tampon, des associations d'ensembles de pages et de pools de mémoire tampon et une commande ALTER SECURITY. Ajoutez l'exemple à la concaténation CSQINP1 de la procédure de tâche démarrée de votre gestionnaire de files d'attente.

**Exemples de fichier CSQINP2**

**Exemple d'objet système CSQ4INSG**

L'exemple de fichier CSQINP2 thlqual.SCSQPROC(CSQ4INSG) contient des définitions des objets système destinés à une utilisation générale suivants :

- Objets par défaut du système
- Objets de commande du système
- Objets d'administration du système
- Autres objets destinés à être utilisés par le système

Vous devez définir les objets dans cet exemple, mais uniquement une fois que le sous-système a démarré. La meilleure façon de procéder consiste à inclure les définitions dans le fichier CSQINP2. Ces définitions sont conservées en cas d'arrêt et de redémarrage du gestionnaire de files d'attente. Vous ne devez pas modifier les noms des objets, mais vous pouvez modifier leurs attributs si nécessaire.

Lorsque les conditions suivantes sont respectées, un message est placé dans la file d'attente SYSTEM.DURABLE.SUBSCRIBER.QUEUE (y compris si la publication/l'abonnement ne sont pas actifs) :

- L'attribut QMGR, PSMODE, est défini sur DISABLED
- L'exemple d'instruction CSQ4INST d'objet DEFINE SUB( ' SYSTEM . DEFAULT . SUB ' ) est présent.

Pour éviter cela, supprimez ou mettez en commentaire l'instruction DEFINE SUB( ' SYSTEM . DEFAULT . SUB ' ) .

La file d'attente de transfert de retard de distribution JMS 2.0 , SYSTEM.DDELAY.LOCAL.QUEUE n'a besoin d'être défini que si le délai de distribution JMS 2.0 est utilisé. Par défaut, la définition de file d'attente est mise en commentaire. Vous pouvez supprimer cette mise en commentaire si nécessaire.

### **Objet système CSQ4INSA et exemple d'authentification**

L'exemple de fichier CSQINP2 thlqual.SCSQPROC(CSQ4INSA) contient la définition de file d'attente du système d'authentification de canal. Cette file d'attente contient les enregistrements des authentifications de canal. Elle contient également les règles d'authentification de canal par défaut.

Vous devez définir les objets de cet exemple si la commande CHLAUTH est activée sur le gestionnaire de files d'attente et si vous souhaitez exécuter des canaux ou si vous souhaitez exécuter la commande SET ou DISPLAY CHLAUTH sur un enregistrement. Il suffit de définir ces objets une fois, lors du premier démarrage du sous-système. La meilleure façon de procéder consiste à inclure les définitions dans le fichier CSQINP2. Ces définitions sont conservées en cas d'arrêt et de redémarrage et vous ne devez donc pas modifier le nom de la file d'attente.

### **Exemple d'objet système CSQ4INSS**

Vous pouvez définir d'autres objets système si vous utilisez des groupes de partage de files d'attente.

L'exemple de fichier thlqual.SCSQPROC(CSQ4INSS) contient des exemples de commande à utiliser avec des structures d'unité de couplage et un ensemble de définitions pour les objets système requis pour les canaux partagés et la mise en file d'attente intra-groupe.

Vous ne devez pas utiliser cet exemple tel quel. Vous devez le personnaliser avant de l'utiliser. Vous pouvez ensuite inclure ce membre dans la concaténation CSQINP2 DD de la procédure de démarrage du gestionnaire de files d'attente ou l'utiliser en tant qu'entrée de la fonction COMMAND de l'utilitaire CSQUTIL afin d'émettre les commandes requises.

Lorsque vous définissez des objets de groupe ou partagés, vous devez les inclure dans la concaténation CSQINP2 DD pour un seul gestionnaire de files d'attente du groupe de partage de files d'attente.

### **Exemple d'objet système CSQ4INSX**

Vous devez définir d'autres objets système si vous utilisez la mise en file d'attente et en cluster partagée.

L'exemple de fichier thlqual.SCSQPROC(CSQ4INSX) contient les définitions de file d'attente requises. Vous pouvez inclure ce membre dans la concaténation CSQINP2 DD de la procédure de démarrage du gestionnaire de files d'attente ou l'utiliser en tant qu'entrée de la fonction COMMAND de l'utilitaire CSQUTIL pour émettre les commandes DEFINE requises.

Il existe deux types de définition d'objet :

- SYSTEM.CHANNEL.xx, requise pour toutes les mises en file d'attente réparties

- SYSTEM.CLUSTER.xx, requise pour la mise en cluster

### **Exemple d'objet JMS système CSQ4INSJ**

Définit les files d'attente utilisées dans le domaine de publication/abonnement JMS.

### **Exemple d'objet système CSQ4INSM**

Si vous utilisez la sécurité de message avancée, vous devez définir des objets système supplémentaires. L'exemple de fichier thlqual.SCSQPROC(CSQ4INSM) contient les définitions de file d'attente requises.

### **Exemple d'objet CSQ4INSR**

Définit les files d'attente utilisées par WebSphere Application Server et les courtiers.

### **Exemple d'objet CSQ4INYD**

Si vous utilisez la mise en file d'attente répartie et que vous deviez configurer vos propres files d'attente, processus et canaux.

L'exemple de fichier thlqual.SCSQPROC(CSQ4INYD) contient des exemples de définition que vous pouvez utiliser pour personnaliser vos objets de mise en file d'attente répartie. Il contient :

- Un ensemble de définitions pour l'extrémité émettrice
- Un ensemble de définitions pour l'extrémité réceptrice
- Un ensemble de définitions pour les clients utilisateur

Vous ne devez pas utiliser cet exemple tel quel. Vous devez le personnaliser avant de l'utiliser. Vous pouvez ensuite inclure ce membre dans la concaténation CSQINP2 DD de la procédure de démarrage du gestionnaire de files d'attente ou l'utiliser en tant qu'entrée de la fonction COMMAND de l'utilitaire CSQUTIL afin d'émettre les commandes DEFINE requises. (Cette méthode est préférable car elle signifie que vous n'êtes pas obligé de redéfinir ces objets à chaque fois que vous redémarrez le gestionnaire de files d'attente).

### **Exemple d'objet CSQ4INYC**

Si vous utilisez la mise en file d'attente, des définitions équivalentes aux définitions de canal et aux définitions de file d'attente éloignée de mise en file d'attente répartie sont créées automatiquement lorsque c'est nécessaire. Toutefois, les définitions de canal manuelles suivantes sont nécessaires : un canal récepteur de cluster pour le cluster et une définition de récepteur de cluster pour l'un au moins des gestionnaires de files d'attente du référentiel.

L'exemple de fichier thlqual.SCSQPROC(CSQ4INYC) contient les exemples de définition suivants, que vous pouvez utiliser pour personnaliser vos objets de cluster :

- Définitions pour le gestionnaire de files d'attente
- Définitions pour le canal récepteur
- Définitions pour le canal émetteur
- Définitions pour les files d'attente de cluster
- Définitions pour les listes de clusters

Vous ne devez pas utiliser cet exemple tel quel. Vous devez le personnaliser avant de l'utiliser. Vous pouvez ensuite inclure ce membre dans la concaténation CSQINP2 DD de la procédure de démarrage du gestionnaire de files d'attente ou l'utiliser en tant qu'entrée de la fonction COMMAND de l'utilitaire CSQUTIL afin d'émettre les commandes DEFINE requises. Cette méthode est préférable car elle signifie que vous n'êtes pas obligé de redéfinir ces objets à chaque fois que vous redémarrez IBM MQ.

### **Exemple d'objet CSQ4INYG**

L'exemple de fichier thlqual.SCSQPROC(CSQ4INYG) contient les exemples de définition suivants, que vous pouvez utiliser pour personnaliser vos propres objets en vue d'une utilisation générale :

- File d'attente de rebut
- File d'attente de transmission par défaut
- Objets d'adaptateur CICS

Vous ne devez pas utiliser cet exemple tel quel. Vous devez le personnaliser avant de l'utiliser. Vous pouvez ensuite inclure ce membre dans la concaténation CSQINP2 DD de la procédure de démarrage du gestionnaire de files d'attente ou l'utiliser en tant qu'entrée de la fonction COMMAND de l'utilitaire CSQUTIL afin d'émettre les commandes DEFINE requises. Cette méthode est préférable car elle signifie que vous n'êtes pas obligé de redéfinir ces objets à chaque fois que vous redémarrez IBM MQ.

Outre les exemples de définition donnés ici, vous pouvez utiliser les définitions d'objet système en tant que base pour vos propres définitions de ressource. Par exemple, vous pouvez faire une copie de travail de SYSTEM.DEFAULT.LOCAL.QUEUE et la renommer MY.DEFAULT.LOCAL.QUEUE. Vous pouvez ensuite modifier les paramètres de cette copie selon vos besoins, puis émettre une commande DEFINE via la méthode de votre choix, à condition que vous ayez les droits nécessaires pour créer des ressources de ce type.

#### **File d'attente de transmission par défaut**

Lisez la description de la [file d'attente de transmission par défaut](#) avant de décider si vous souhaitez définir une nouvelle file d'attente de ce type.

- Si vous décidez de définir une nouvelle file d'attente de transmission par défaut, n'oubliez pas que vous devez également définir un canal pour la gérer.
- Si vous décidez de ne pas définir cette nouvelle file d'attente, n'oubliez pas de supprimer l'instruction DEFXMLTQ de la commande ALTER QMGR de l'exemple.

#### **Objets d'adaptateur CICS**

L'exemple définit une file d'attente d'initiation nommée CICS01.INITQ. Celle-ci est utilisée par la transaction CKTI fournie par IBM MQ. Vous pouvez modifier le nom de cette file d'attente. Ce nom doit toutefois correspondre au nom défini dans la table d'initialisation du système CICS ou au remplacement SYSIN de l'instruction INITPAR.

#### **Exemples d'objet CSQ4INYS/CSQ4INYR**

Définitions de classe d'archivage permettant d'utiliser :

- un ensemble de pages pour chaque classe de message
- plusieurs ensembles de pages pour les principales classes de message

Par exemple, SYSTEM.COMMAND.INPUT utilise STGCLASS ('SYSVOLAT') et SYSTEM.CLUSTER.TRANSMIT.QUEUE utilise STGCLASS ('REMOTE'). Dans CSQ4INYS, ces deux classes de stockage utilisent le même ensemble de pages. Dans CSQ4INYR, ces classes de stockage utilisent des ensembles de pages différents afin de réduire l'impact du remplissage de la file d'attente de transmission.

#### **Exemples CSQINPT**

##### **CSQ4INST**

L'exemple de fichier thlqual.SCSQPROC(CSQ4INST) contient la définition de l'abonnement par défaut du système.

Vous devez définir l'objet dans cet exemple, et ce une seule fois lors du premier démarrage du moteur de publication/abonnement. La meilleure façon de procéder consiste à inclure la définition dans le fichier CSQINPT. Cette définition est conservée en cas d'arrêt et de redémarrage du gestionnaire de files d'attente. Vous ne devez pas modifier le nom de l'objet, mais vous pouvez modifier leurs attributs si nécessaire.

##### **CSQ4INYT**

Le fichier thlqual.SCSQPROC(CSQ4INYT) contient un ensemble de commandes que vous pouvez exécuter lors du démarrage du moteur de publication/abonnement. Cet exemple affiche des informations sur les rubriques et les abonnements.

## Autre

### Exemple d'affichage CSQ4DISP

L'exemple de fichier thlqual.SCSQPROC(CSQ4DISP) contient un ensemble de commandes DISPLAY génériques affichant toutes les ressources définies dans votre gestionnaire de files d'attente, notamment les définitions de tous les objets et définitions IBM MQ tels que les classes d'archivage et la trace. Ces commandes peuvent générer une sortie très volumineuse. Vous pouvez utiliser cet exemple dans le fichier CSQINP2 ou en tant qu'entrée de la fonction COMMAND de l'utilitaire CSQUTIL.

### Exemple de fichier CSQ4INPX

Le fichier thlqual.SCSQPROC(CSQ4INPX) contient un ensemble de commandes que vous souhaitez peut-être exécuter à chaque démarrage de l'initiateur de canal. Vous devez personnaliser l'exemple pour pouvoir l'utiliser. Vous pouvez ensuite l'inclure dans le fichier CSQINPX de l'initiateur de canal.

### Exemples de fichier CSQ4IVPQ et CSQ4IVPG

Les exemples de fichier thlqual.SCSQPROC(CSQ4IVPQ) et thlqual.SCSQPROC(CSQ4IVPG) contiennent des ensembles de commandes DEFINE nécessaires à l'exécution des programmes de vérification de l'installation.

Vous pouvez inclure ces exemples au fichier CSQINP2. Une fois ces programmes exécutés, il n'est plus nécessaire de les exécuter à nouveau à chaque redémarrage du gestionnaire de files d'attente. Il est donc inutile de conserver en permanence ces exemples dans la concaténation CSQINP2.

### Exemples CSQ4MSTR et CSQ4MSRR

thlqual.SCSQPROC(CSQ4MSTR) et thlqual.SCSQPROC(CSQ4MSRR) sont des exemples de procédure de tâche démarrée pour le gestionnaire de files d'attente.

CSQ4MSRR utilise CSQ4INYR dans la concaténation CSQINP2 de sorte que les files d'attente importantes soient réparties sur différents ensembles de pages.

Vous pouvez supprimer les commentaires afin d'utiliser la carte CSQMINS pour les gestionnaires de files d'attente créés, si nécessaire.

### Exemple CSQ4QMIN

thlqual.SCSQPROC(CSQ4QMIN) est un exemple d'ensemble de données QMINI.

Voir [Fichier QMINI](#) pour des détails sur le fichier QMINI et la strophe **TransportSecurity**.

## Reprise et redémarrage sur z/OS

Utilisez les liens contenus dans cette rubrique pour en savoir plus sur les fonctions de redémarrage et de reprise d'IBM MQ for z/OS.

IBM MQ for z/OS est doté de fonctions robustes de redémarrage et de reprise. Pour en savoir plus sur le redémarrage d'un gestionnaire de files d'attente après son arrêt et sur les événements liés à ce redémarrage, consultez les rubriques secondaires ci-après.

- [«Comment les données sont-elles modifiées dans IBM MQ for z/OS ?»](#), à la page 271
- [«Conservation de la cohérence dans IBM MQ for z/OS»](#), à la page 272
- [«Que se passe-t-il lors de l'arrêt dans IBM MQ for z/OS ?»](#), à la page 275
- [«Que se passe-t-il lors du redémarrage et de la reprise dans IBM MQ for z/OS ?»](#), à la page 276
- [«Comment sont résolues les unités de récupération en attente de validation»](#), à la page 279
- [«Récupération d'une file d'attente partagée»](#), à la page 281

## Concepts associés

► **z/OS** [Actions de reprise IBM MQ for z/OS](#)

[Sources à partir desquelles vous pouvez émettre des commandes MQSC sous z/OS](#)

## Tâches associées

[Planification de la sauvegarde et de la récupération](#)

► **z/OS** [Administration de z/OS](#)

## Référence associée

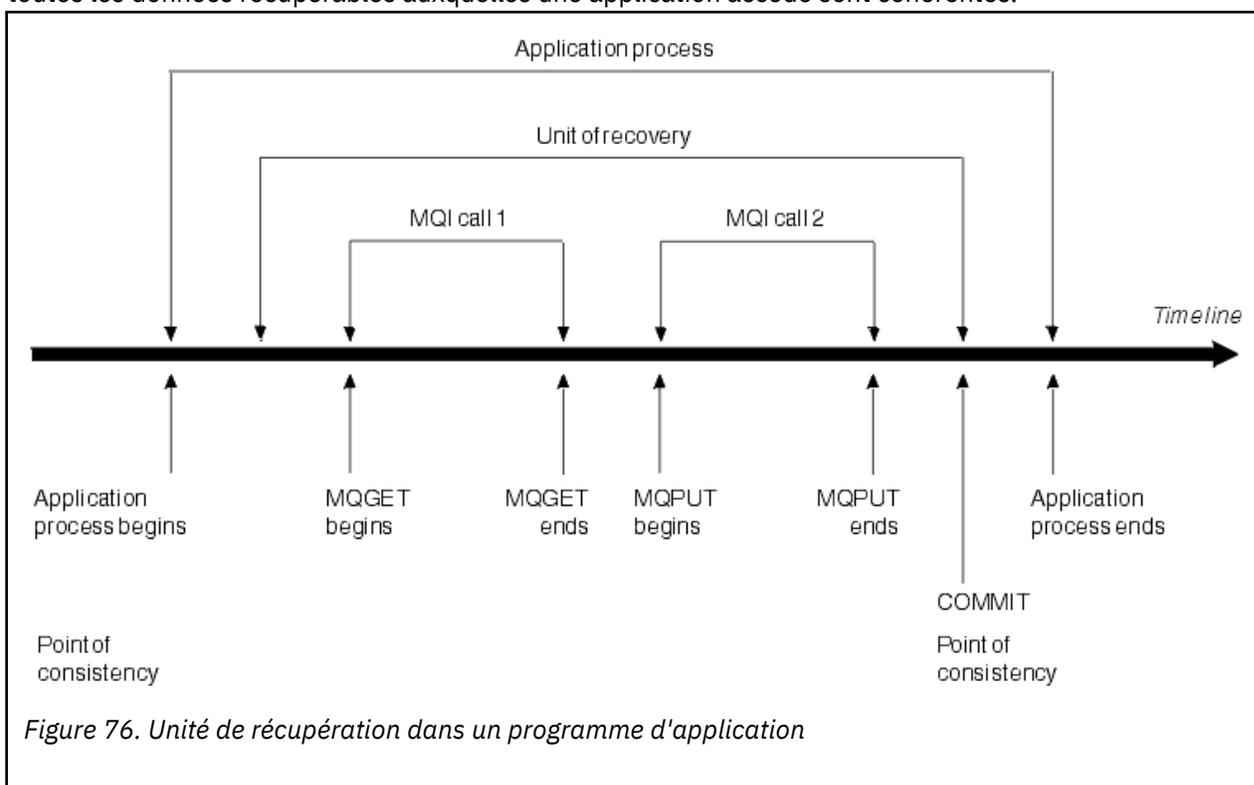
► **z/OS** [Messages pour IBM MQ for z/OS](#)

## ► **z/OS** Comment les données sont-elles modifiées dans IBM MQ for z/OS ?

IBM MQ for z/OS doit interagir avec d'autres sous-systèmes pour garantir la cohérence de toutes les données. Cette rubrique contient des informations sur les *unités de récupération*, leur nature et leur utilisation lors d'une *annulation*.

## Unités de récupération

Une *unité de récupération* est le traitement effectué par un gestionnaire de files d'attente unique pour un programme d'application, qui modifie les données IBM MQ d'un point de cohérence vers un autre. Le *point de cohérence* (également appelé *point de synchronisation* ou *point de validation*) est l'instant où toutes les données récupérables auxquelles une application accède sont cohérentes.



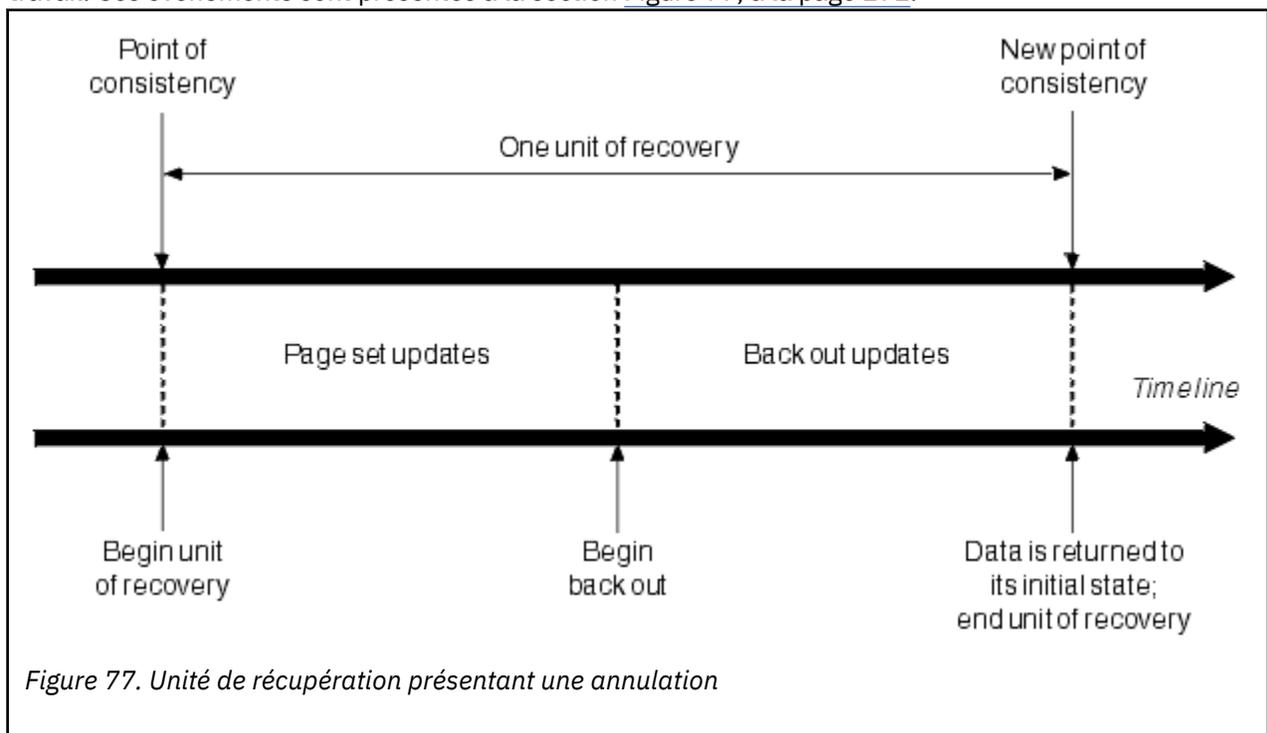
Une unité de récupération commence à la première modification apportée aux données après le lancement du programme ou après le point de cohérence précédent. Elle se termine par le point de cohérence suivant. [Figure 76](#), à la [page 271](#) présente la relation entre les unités de récupération, le point de cohérence et un programme d'application. Dans cet exemple, le programme d'application apporte des modifications aux files d'attente via les appels MQI 1 et 2. Le programme d'application peut inclure plusieurs unités de récupération ou un seul. Toutefois, toutes les unités de récupération terminées se finissent par un point de validation.

Par exemple, une transaction bancaire permet de transférer des fonds d'un compte à un autre. Tout d'abord, le programme soustrait le montant du premier compte, le compte A. Ensuite, il ajoute le montant au deuxième compte, B. Après avoir soustrait le montant de A, les deux comptes sont incohérents et IBM MQ ne peut pas valider. Elles deviennent cohérentes lorsque le montant est ajouté au compte B. Lorsque les deux étapes sont terminées, le programme peut annoncer un point de cohérence au moyen d'une validation, ce qui rend les modifications visibles des autres programmes d'application.

L'arrêt normal d'un programme d'application entraîne automatiquement un point de cohérence. Certaines demandes de programme dans les programmes CICS et IMS génèrent également un point de cohérence, par exemple, EXEC CICS SYNCPOINT.

## Annulation d'un travail

Si une erreur se produit dans une unité de récupération, IBM MQ supprime les modifications des données en rétablissant leur état qui existait au début de l'unité de récupération, à savoir qu'IBM MQ annule le travail. Ces événements sont présentés à la section Figure 77, à la page 272.



## z/OS Conservation de la cohérence dans IBM MQ for z/OS

Les données dans IBM MQ for z/OS doivent être cohérentes avec Batch, CICS, IMS ou TSO. Les données modifiées dans l'une de ces applications doivent l'être également dans l'autre.

Avant qu'un système ne valide les données modifiées, il doit savoir que l'autre système peut appliquer les modifications correspondantes. Les deux systèmes doivent donc communiquer.

Pendant une *validation en deux phases* (par exemple, sous CICS), un sous-système coordonne le processus. Ce sous-système est appelé le *coordinateur*. L'autre est le *participant*. CICS ou IMS est toujours le coordinateur dans les interactions avec IBM MQ, et IBM MQ est toujours le participant. Dans l'environnement Batch ou TSO, IBM MQ peut participer dans les protocoles de validation en deux phases coordonnés par z/OS RRS.

Pendant une *validation en une seule phase* (par exemple, sous TSO ou Batch), IBM MQ est toujours le coordinateur dans les interactions et contrôle complètement la validation.

Dans un environnement WebSphere Application Server, la sémantique de l'objet de session JMS détermine si la coordination de la validation en deux phases ou une seule phase est utilisée.

## **Cohérence avec CICS ou IMS**

La connexion entre IBM MQ et CICS ou IMS prend en charge les protocoles de point de synchronisation suivants :

- Validation en deux phases : pour les transactions qui mettent à jour les ressources appartenant à plusieurs gestionnaires de ressources.

Il s'agit du protocole de point de synchronisation réparti standard. Il suppose une plus grande journalisation et davantage de flux de messages que la validation en une phase.

- Validation en une seule phase : pour les transactions qui mettent à jour les ressources détenues par un seul gestionnaire de ressources (IBM MQ).

Ce protocole est optimisé pour la journalisation et les flux de messages.

- Non prise en compte de point de synchronisation : pour les transactions qui impliquent IBM MQ, mais qui n'exécutent aucune opération dans le gestionnaire de files d'attente qui nécessite un point de synchronisation (par exemple, consultation d'une file d'attente).

Dans chaque cas, CICS ou IMS fait office de gestionnaire de points de synchronisation.

Les étapes de la validation en deux phases qu'utilise IBM MQ pour communiquer avec CICS ou IMS sont les suivantes :

1. Lors de la phase 1, chaque système détermine indépendamment s'il a enregistré suffisamment d'informations dans son journal et s'il peut valider les tâches exécutées.

A la fin de cette phase, les systèmes communiquent. S'ils sont d'accord, la phase suivante commence.

2. Lors de la phase 2, les modifications deviennent permanentes. Si l'un des deux systèmes s'arrête de manière anormale au cours de cette phase, l'opération se termine par le processus de reprise lors du redémarrage.

### **Illustration du processus de validation en deux phases**

Figure 78, à la page 274 illustre le processus de validation en deux phases. Les événements dans le coordinateur CICS ou IMS sont indiqués sur la ligne supérieure. Les événements dans IBM MQ se trouvent sur la ligne du bas.

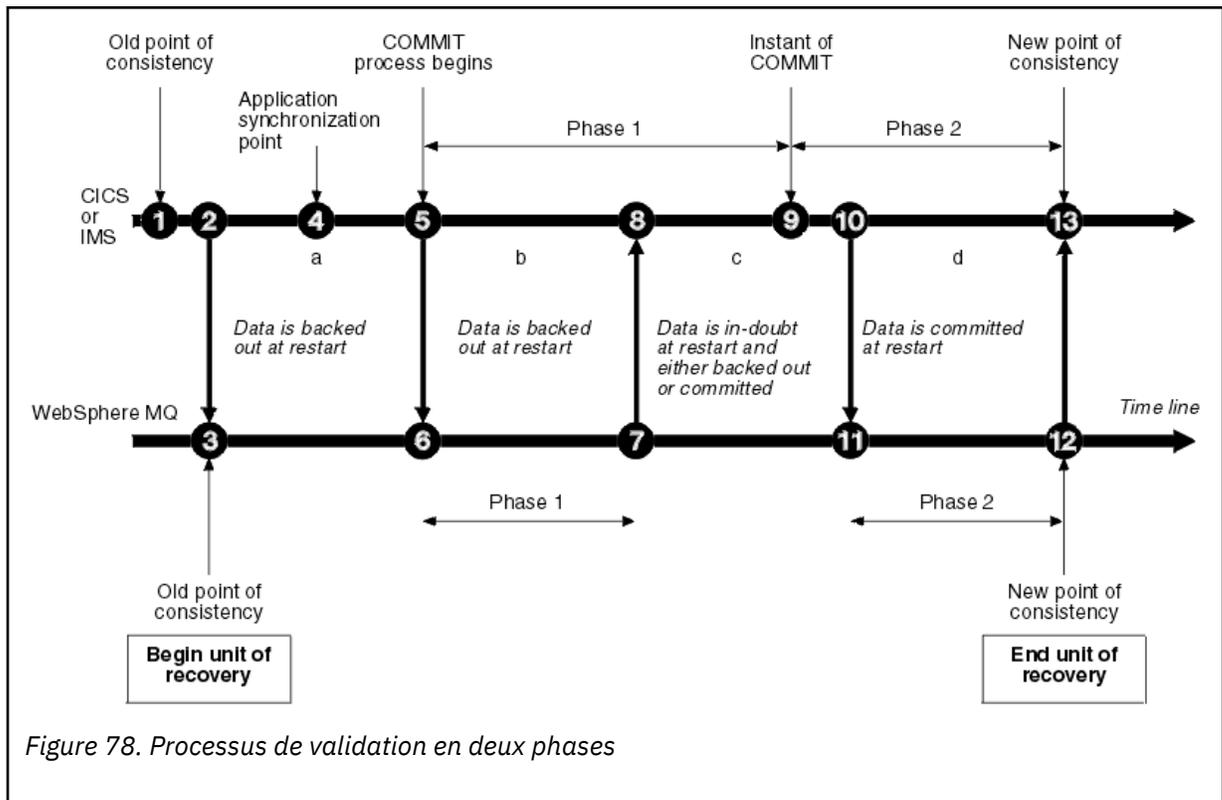


Figure 78. Processus de validation en deux phases

Les nombres cités dans la section suivante font référence aux nombres indiqués dans la figure.

1. Les données contenues dans le coordinateur sont à leur point de cohérence.
2. Un programme d'application du coordinateur appelle IBM MQ pour mettre une file d'attente à jour en ajoutant un message.
3. Cette opération démarre une unité de récupération dans IBM MQ.
4. Le traitement se poursuit jusqu'à ce qu'un point de synchronisation soit atteint.
5. Le coordinateur démarre le traitement de la validation. Les programmes CICS utilisent une commande SYNCPOINT ou une fin normale d'application pour démarrer la validation. Les programmes IMS peuvent lancer la validation en utilisant un appel CHKP, SYNC, GET UNIQUE vers l'IOPCB, ou un arrêt normal d'application. La phase 1 du processus de validation commence.
6. Lorsque le coordinateur démarre le traitement de la phase 1, IBM MQ en fait de même.
7. IBM MQ exécute correctement la phase 1, consigne cette donnée dans son journal et notifie le coordinateur.
8. Le coordinateur reçoit la notification.
9. Le coordinateur termine le traitement de la phase 1. Les deux sous-systèmes acceptent de valider les modifications, car tous deux ont terminé la phase 1 et peuvent procéder à une reprise en cas d'erreur. Le coordinateur enregistre le moment de la validation, c'est-à-dire la décision irrévocable d'appliquer les modifications, dans son journal.  
Le coordinateur démarre la phase 2 du traitement, c'est-à-dire la validation en tant que telle.
10. Le coordinateur notifie IBM MQ de démarrer la phase 2.
11. IBM MQ consigne le début de la phase 2.
12. La phase 2 s'est terminée correctement. Elle constitue désormais un nouveau point de cohérence IBM MQ. IBM MQ notifie alors le coordinateur que le traitement de la phase 2 est terminé.
13. Le coordinateur termine le traitement de la phase 2. Les données contrôlées par les deux sous-systèmes sont désormais cohérentes et disponibles pour les autres applications.

## Gestion de la cohérence après un arrêt anormal

Lorsqu'un gestionnaire de files d'attente redémarre après un arrêt anormal, il doit déterminer s'il valide ou annule les unités de récupération qui étaient actives lors de l'arrêt. Pour certaines unités de récupération, IBM MQ dispose d'informations suffisantes pour prendre la décision. Pour d'autres, ce n'est pas le cas et le produit doit obtenir des informations de la part du coordinateur lors du rétablissement de la connexion.

Figure 78, à la page 274 indique quatre périodes au cours des deux phases : a, b, c et d. Le statut d'une unité de récupération dépend de la période pendant laquelle la résiliation s'est produite. Ce statut peut être l'un des suivants :

### En cours

Le gestionnaire de files d'attente s'est arrêté avant la fin de la phase 1 (période a ou b). Lors du redémarrage IBM MQ annule les mises à jour.

### Indéfini

Le gestionnaire de files d'attente s'est arrêté après avoir terminé la phase 1 et avant de démarrer la phase 2 (période c). Seul le coordinateur sait si l'erreur s'est produite avant ou après la validation (point 9). Si elle s'est produite avant, IBM MQ doit annuler les modifications. Si elle s'est produite après, IBM MQ doit exécuter les modifications et les valider. Lors du redémarrage, IBM MQ attend d'autres informations du coordinateur avant de traiter cette unité de récupération.

### En cours de validation

Le gestionnaire de files d'attente s'est arrêté après avoir commencé le traitement de la phase 2 (période d). Il exécute les modifications validées.

### En cours d'annulation

Le gestionnaire de files d'attente s'est arrêté après le début de l'annulation d'une unité de récupération, mais avant la fin de ce processus (n'apparaît pas dans la figure). Lors du redémarrage IBM MQ poursuit l'annulation des modifications.

## ► z/OS Que se passe-t-il lors de l'arrêt dans IBM MQ for z/OS ?

Un gestionnaire de files d'attente s'arrête normalement en réponse à la commande STOP QMGR. S'il s'arrête pour une autre raison, cet arrêt est anormal.

Notez que pendant l'arrêt du gestionnaire de files d'attente IBM MQ émet en interne la commande

```
DISPLAY CONN(*) TYPE(CONN) ALL WHERE (APPLTYPE NE SYSTEMAL)
```

afin que vous sachiez quelle unité d'exécution risque de bloquer l'arrêt normal du gestionnaire de files d'attente.

SYSTEMAL correspond à APPLTYPES de SYSTEM ou CHINIT ; ainsi, la commande DISPLAY CONN, qui filtre les types d'application ne correspondant pas à SYSTEMAL, renvoie au journal de travail les informations relatives aux unités d'exécution qui pourraient bloquer l'arrêt normal.

### Arrêt normal

Dans un arrêt normal, IBM MQ cesse toute activité de manière ordonnée. Vous pouvez arrêter IBM MQ à l'aide du mode repos, du mode forcé ou du mode de redémarrage. Les conséquences sont décrites à la section [Tableau 22](#), à la page 275.

Type d'unité d'exécution	QUIESCE	FORCE	RESTART
Unités d'exécution actives	Exécution jusqu'à la fin	Renvoyer	Renvoyer
Nouvelles unités d'exécution	Peut démarrer	Non autorisé	Non autorisé

Tableau 22. Arrêt à l'aide des commandes QUIESCE, FORCE et RESTART (suite)			
Type d'unité d'exécution	QUIESCE	FORCE	RESTART
Nouvelles connexions	Non autorisé	Non autorisé	Non autorisé

Les applications par lots sont notifiées si un arrêt se produit alors que l'application est toujours connectée.

Avec CICS, une unité d'exécution en cours s'exécute uniquement jusqu'à la fin de l'unité de récupération. Avec CICS, l'arrêt d'un gestionnaire de files d'attente en mode repos arrête l'adaptateur CICS, si bien que si une tâche active contient plusieurs unités de récupération, elle ne s'exécute pas nécessairement complètement.

Si vous arrêtez un gestionnaire de files d'attente en mode forcé ou en mode de redémarrage, aucune nouvelle unité d'exécution n'est allouée et le travail en cours sur les unités d'exécution connectées est annulé. L'utilisation de ces modes peut créer des unités de récupération en attente de validation pour les unités d'exécution se trouvant entre des phases de traitement de validation. Elles sont résolues lorsqu'IBM MQ se reconnecte au sous-système CICS, IMS ou RRS.

Lorsque vous arrêtez un gestionnaire de files d'attente, les étapes sont les suivantes, quel que soit le mode utilisé :

1. Les connexions sont interrompues.
2. IBM MQ n'accepte plus les commandes.
3. IBM MQ vérifie que les mises à jour restantes apportées aux ensembles de pages sont terminées.
4. La commande DISPLAY USAGE est émise en interne par IBM MQ de sorte que l'adresse de redémarrage relative en octets est enregistrée dans le journal de la console z/OS.
5. Le point de contrôle du redémarrage est pris en compte et le fichier d'amorce est mis à jour.

Les arrêts qui définissent le mode repos n'ont aucune incidence sur les unités de récupération en attente de validation. Les unités en attente de validation le restent.

### Arrêt anormal

Un arrêt anormal peut laisser certaines données dans un état incohérent, par exemple :

- Une unité de récupération a été interrompue avant d'atteindre un point de cohérence.
- Les données validées n'ont pas été écrites dans les ensembles de pages.
- Les données non validées ont été écrites dans les ensembles de pages.
- Un programme d'application a été interrompu entre la phase 1 et la phase 2 du processus de validation. L'unité de récupération est alors en attente de validation.

IBM MQ résout les incohérences de données liées à un arrêt anormal lors du redémarrage et de la récupération.

## Que se passe-t-il lors du redémarrage et de la reprise dans IBM MQ for z/OS ?

IBM MQ utilise son journal de reprise et le fichier d'amorce pour déterminer les données à récupérer lors de son redémarrage. Le fichier d'amorce identifie le jeu de données de journal actif et le fichier journal d'archivage, ainsi que l'emplacement du point de cohérence IBM MQ dans le journal.

### Présentation du redémarrage et de la reprise

Une fois IBM MQ initialisé, le processus de redémarrage du gestionnaire de files d'attente se déroule comme suit :

- Initialisation du journal

- Recréation du statut en cours
- Reprise par lecture vers l'aval
- Reprise par lecture vers l'amont
- Régénération de l'index des files d'attente

Une fois la reprise terminée :

- Les modifications validées sont reflétées dans les données.
- L'activité en attente de validation est reflétée dans les données. Toutefois, ces données sont verrouillées et inutilisables tant qu'IBM MQ n'a pas reconnu la décision en attente de validation et agit sur celle-ci.
- Les modifications en cours et en cours de suppression qui ont été interrompues ont été supprimées des files d'attente. Les messages sont cohérents et peuvent être utilisés.
- Un nouveau point de contrôle est pris en compte.
- De nouveaux index ont été générés pour les files d'attente indexées contenant des messages persistants (décrits à la section «Régénération des index de files d'attente», à la page 278).

Si des fichiers d'amorce en double sont utilisés, IBM MQ vérifie la cohérence des horodatages de ces fichiers :

- Si les deux copies du fichier d'amorce sont actuelles, IBM MQ vérifie que les deux horodatages sont identiques. S'ils ne le sont pas, IBM MQ émet le message CSQJ120E et s'arrête. Ceci peut se produire lorsque les deux copies du fichier d'amorce sont gérées sur des volumes DASD distincts et que l'un des volumes a été restauré alors que le gestionnaire de files d'attente était arrêté. IBM MQ détecte la situation au redémarrage.
- Si une copie du fichier d'amorce a été libérée et que la journalisation se poursuit avec un seul fichier d'amorce, un problème peut survenir. Si les deux copies sont gérées sur le même volume et que celui-ci a été restauré ou que les deux copies ont été restaurées séparément, IBM MQ risque de ne pas détecter la restauration. Dans ce cas, les enregistrements de journal non consignés dans le fichier d'amorce sont inconnus du système.

Les applications par lots ne sont pas notifiées lorsque le redémarrage se produit *après* que l'application a demandé une connexion.

## Présentation de la plage de données de journal requise pour la reprise

Lors du redémarrage, la plage des données de journal devant être lue dépend de plusieurs facteurs :

- Lors d'une fin anormale, le système contient en général plusieurs unités d'oeuvre non terminées. Comme décrit ci-dessus, le processus de redémarrage permet d'amener le système à un état de cohérence pouvant impliquer l'annulation des unités d'oeuvre en cours ou la récupération des verrous existant sur les unités d'oeuvre en attente de validation. La récupération des unités d'oeuvre requiert que tous les enregistrements de journal des unités d'oeuvre en cours, en cours d'annulation et en attente de validation soient disponibles. IBM MQ court-circuite les anciennes unités d'oeuvre. La reprise est ainsi exécutée sur une plage de données de journal moins étendue.
- Lors d'un arrêt anormal, de nombreuses mises à jour persistantes sont contenues uniquement dans la mémoire cache du pool de mémoire tampon. Elles n'ont pas encore été écrites sur le disque. Ces modifications doivent être lues à partir du journal, puis appliquées de nouveau aux données contenues dans les ensembles de pages. Les adresses relatives en octets de récupération des ensembles de pages dans le point de cohérence décrivent l'adresse relative en octets de journal la plus basse requise pour faire passer les ensembles de pages à un état cohérent.
- Si d'anciens ensembles de pages ont été introduits dans le système, par exemple si une sauvegarde d'ensemble de pages a été introduite en vue d'une reprise après la défaillance d'un support, toutes les modifications doivent être lues dans le journal à partir du moment où la sauvegarde a été effectuée. Ces modifications sont alors appliquées à nouveau aux données contenues dans l'ensemble de pages en cours de récupération. Les adresses relatives en octets de la récupération des ensembles de pages

contenues dans la page 0 de l'ensemble de pages décrivent l'adresse de journal la plus basse pour la reprise d'un ensemble de pages.

- Si vous utilisez des messages persistants sur les files d'attente partagées, une plage de données de journal est requise pour récupérer les CFSTRUCT contenant ces messages. Les données de journal les plus anciennes nécessaires à la reprise CFSTRUCT datent environ de l'heure de l'ancienne commande CFSTRUCT BACKUP.

Dans le cadre de l'exécution normale, la commande DISPLAY USAGE TYPE(DATASET) peut être utilisée pour afficher la plage du journal de reprise associée à ces facteurs (elle est évidemment incapable de donner des informations en raison de la réintroduction des anciens ensembles de pages). Pour éviter les problèmes risquant de prolonger le processus de redémarrage d'un gestionnaire de files d'attente dans le cas d'un arrêt anormal, surveillez de manière régulière les valeurs obtenues par la commande DISPLAY USAGE TYPE(DATASET).

En outre, le gestionnaire de files d'attente émet des messages d'information relatifs à ces facteurs :

- CSQJ160I et CSQJ161I : avertissent de la présence d'unités d'oeuvre à exécution longue.
- CSQR026I et CSQR027I : indiquent si ces unités d'oeuvre à exécution longue ont été court-circuitées.
- CSQE040I et CSQE041E : avertissent que les sauvegardes de structure deviennent anciennes et qu'une opération RECOVER CFSTRUCT demanderait beaucoup de temps.

## **Identification de l'application pour laquelle une unité d'oeuvre à exécution longue existe**

Il est possible d'identifier l'application pour laquelle une unité d'oeuvre à exécution longue existe. Pour cela, vous utilisez la commande DISPLAY CONN.

Cette commande renvoie des informations de connexion sur toutes les applications connectées au gestionnaire de files d'attente ainsi que des informations supplémentaires vous aidant à identifier les applications pour lesquelles une unité d'oeuvre à exécution longue existe. Les informations renvoyées par la commande DISPLAY CONN sont semblables aux informations renvoyées par la commande DISPLAY QSTATUS, la différence principale étant que la commande DISPLAY CONN affiche des informations sur les objets et des informations transactionnelles sur une connexion donnée, plutôt que des détails sur les connexions associées à un objet donné.

Pour chaque application connectée, la commande DISPLAY CONN renvoie les informations suivantes :

- Informations de base, notamment l'ID connexion et l'ID processus.
- Informations transactionnelles relatives à cette connexion, notamment l'heure et la date de création de la transaction (c'est-à-dire de la première émission de la commande MQGET/PUT sous le contrôle d'un point de synchronisation) et de première écriture de la transaction dans le journal.
- Informations horaires indiquant pour quelle application une unité d'oeuvre à exécution longue est toujours en cours d'exécution.
- Liste de tous les objets actuellement ouverts par la connexion. Les détails relatifs à chaque objet sont renvoyés dans un message distinct, l'ID connexion étant utilisé en tant que clé. Il existe différents types d'objet, par exemple les files d'attente et les gestionnaires de files d'attente. Les informations affichées sont donc propres à chacun de ces types.

## **Régénération des index de files d'attente**

Pour accélérer les opérations MQGET sur une file d'attente où les messages ne sont pas extraits de manière séquentielle, vous pouvez spécifier que vous souhaitez qu'IBM MQ gère un index des identificateurs de message ou de corrélation ou un ID groupe pour tous les messages de cette file d'attente.

Lorsqu'un gestionnaire de files d'attente redémarre, ces index sont régénérés pour chaque file d'attente. Cette opération s'applique uniquement aux messages persistants. Les messages non persistants sont

supprimés lors du redémarrage. Si les files d'attente indexées contiennent un grand nombre de messages persistants, le temps nécessaire au redémarrage du gestionnaire de files d'attente augmente.

Vous pouvez choisir de régénérer les index de manière asynchrone lors du démarrage du gestionnaire de files d'attente à l'aide du paramètre QINDEXBLD de la macro CSQ6SYSP. Si vous définissez QINDEXBLD=NOWAIT, IBM MQ redémarre sans attendre que les index soient régénérés.

## Comment sont résolues les unités de récupération en attente de validation

Si IBM MQ perd sa connexion à un autre gestionnaire de ressources, il tente généralement de récupérer tous les objets incohérents au redémarrage.

Si IBM MQ perd sa connexion à CICS, IMS, ou RRS, il tente généralement de récupérer tous les objets incohérents au redémarrage. Les informations requises pour résoudre les unités de récupération en attente doivent provenir du système de coordination. Les sections suivantes décrivent le processus de résolution pour différents environnements.

- [Comment sont résolues les unités de récupération en attente de validation à partir de CICS](#)
- [Comment sont résolues les unités de récupération en attente de validation à partir de IMS](#)
- [Comment sont résolues les unités de récupération en attente de validation à partir de RRS](#)
- [Comment sont résolues les unités de récupération en attente de validation avec une disposition des unités de récupération GROUPE](#)

### Résolution des unités de récupération en attente de validation depuis CICS

Dans certains cas, CICS ne peut pas exécuter le processus IBM MQ pour résoudre les unités de récupération en attente de validation. Dans ce cas, IBM MQ envoie les messages suivants :

- CSQC404E
- CSQC405E
- CSQC406E
- CSQC407E

suivi du message CSQC408I.

Pour plus de détails sur la signification de ces messages, voir le manuel [Messages, codes achèvement et codes anomalie IBM MQ for z/OS](#).

La résolution d'unités en attente de validation n'a pas d'impact sur les ressources CICS. CICS contrôle la coordination de la récupération et valide ou annule automatiquement chaque unité au redémarrage selon qu'il existe un enregistrement de journal ayant marqué le début de la validation. L'existence d'objets en attente de validation ne verrouille pas les ressources CICS lors de la reconnexion d'IBM MQ.

L'une des fonctions de l'adaptateur CICS est de maintenir les données synchronisées entre CICS et IBM MQ. Si un gestionnaire de files d'attente s'arrête anormalement alors qu'il est connecté à CICS, CICS peut valider ou annuler le travail sans qu'IBM MQ le sache. Lorsque le gestionnaire de files d'attente redémarre, le travail est qualifié *en attente de validation*.

IBM MQ ne peut pas résoudre ces unités de récupération en attente de validation (c'est à dire, valider ou annuler les changements apportés aux ressources IBM MQ) tant que la connexion à CICS n'est pas redémarrée et rétablie.

Un processus de résolution des unités de récupération en attente de validation est initié au cours du redémarrage de l'adaptateur CICS. Le processus démarre lorsque l'adaptateur demande une liste d'unités de récupération en attente de validation. Ensuite :

- L'adaptateur extrait la liste des unités de récupération en attente de validation de l'ID de connexion depuis IBM MQ et les envoie à CICS pour qu'il les résolve.

- CICS compare les entrées de la liste aux entrées dans son journal. CICS détermine dans sa liste l'action qu'il a exécuté pour chaque unité de récupération en attente de validation.

Pour toutes les unités résolues, IBM MQ met à jour les files d'attente lorsque cela est nécessaire et publie les verrous correspondants. Des unités non résolues peuvent rester après le redémarrage. Résolvez-les en suivant les méthodes décrites dans [Administration de IBM MQ for z/OS](#).

## Résolution des unités de récupération en attente de validation depuis IMS

La résolution des unités de récupération en attente de validation dans IMS n'affecte pas les ressources DL/I. IMS contrôle la coordination de la récupération et, au redémarrage, valide ou annule automatiquement le travail Data Language/I incomplet. La décision de validation ou d'annulation pour les régions en ligne (chemin non rapide) dépend de la présence ou de l'absence de types d'enregistrement de journal IMS X'3730' et X'3801'. L'existence d'unités de récupération en attente de validation n'implique pas que les enregistrements Data Language/I soient verrouillés jusqu'à ce que IBM MQ se connecte.

Lors du redémarrage du gestionnaire de files d'attente, IBM MQ établit la liste des unités de récupération en attente de validation. IMS crée sa propre liste d'entrées de récupération résiduelles (RRE). Les entrées de récupération résiduelles sont consignées dans des points de récupération IMS jusqu'à ce que toutes les entrées soient résolues.

Lors de la reconnexion d'une région IMS à IBM MQ, IMS indique à IBM MQ s'il doit valider ou annuler les unités de récupération marquées dans IBM MQ comme étant en attente de validation.

Lorsque les unités en attente de validation sont résolues :

1. Si IBM MQ détermine qu'il a marqué une entrée pour la validation et que IMS l'a marquée pour l'annuler, IBM MQ génère un message CSQQ010E. IBM MQ génère ce message pour toutes les incohérences de ce type entre IBM MQ et IMS.
2. Si IBM MQ dispose d'unités en attente de validation, l'adaptateur envoie le message CSQQ008I.

Pour toutes les unités résolues, IBM MQ met à jour les files d'attente lorsque cela est nécessaire et libère les verrous correspondants.

IBM MQ maintient les verrous sur le travail en attente de validation qui n'a pas été résolu. Cela peut entraîner un retard dans le système si des verrous importants sont mis en attente. La connexion reste active afin de pouvoir résoudre les entrées de récupération résiduelles IMS. Récupérez les unités d'exécution en attente de validation en suivant les méthodes décrites dans [Administration de IBM MQ for z/OS](#).

Tout le travail en attente de validation doit être résolu sauf en cas de problèmes logiciels ou de fonctionnement, comme, par exemple, dans le cas d'un démarrage à froid d'IMS. La résolution en attente de validation par la région de contrôle IMS s'effectue dans deux cas :

1. Au début de la connexion à IBM MQ pendant laquelle la résolution s'effectue de façon synchrone
2. Lorsqu'un programme s'arrête anormalement, la résolution s'effectue de façon asynchrone.

## Comment sont résolues les unités de récupération en attente de validation à partir de RRS

L'une des fonctions de l'adaptateur RRS est de maintenir les données synchronisées entre IBM MQ et les autres gestionnaires de ressources impliqués RRS. Si un problème apparaît lorsqu'IBM MQ a terminé la première phase de la validation et attend une décision de RRS (le coordinateur de validation), l'unité de récupération passe en attente de validation.

Lorsque la communication est rétablie entre RRS et IBM MQ, RRS valide ou annule automatique chaque unité de récupération, en fonction de la présence d'un enregistrement de journal marquant le début de la validation. IBM MQ ne peut pas résoudre ces unités de récupération en attente de validation (c'est à dire, valider ou annuler les changements apportés aux ressources IBM MQ) tant que la connexion à RRS n'est pas rétablie.

D Dans certains cas, RRS ne peut résoudre les unités de récupération en attente de validation. Dans ce cas, IBM MQ envoie l'un des messages suivants à la console z/OS :

- CSQ3011I
- CSQ3013I
- CSQ3014I
- CSQ3016I

Pour plus de détails sur la signification de ces messages, voir le manuel [Messages, codes achèvement et codes anomalie IBM MQ for z/OS](#).

Pour toutes les unités résolues, IBM MQ met à jour les files d'attente lorsque cela est nécessaire et lève les verrous correspondants. Les unités de récupération non résolues peuvent rester après le redémarrage. Résolez-les en suivant les méthodes décrites dans [Administration de IBM MQ for z/OS](#).

## **Comment sont résolues les unités de récupération en attente de validation avec une disposition des unités de récupération GROUPE**

Les transactions en attente de validation qui ont une disposition d'unités de récupération GROUPE peuvent être résolues par le coordinateur de transactions par un gestionnaire de files d'attente dans le groupe de partage de files d'attente lorsque l'attribut de gestionnaire de files d'attente GROUPUR est activé. Lorsqu'un coordinateur de transactions se connecte à nouveau, il demande généralement une liste des transactions en attente de validation puis les résout à l'aide des informations provenant de ses journaux.

Lorsqu'un coordinateur de transactions, connecté à une disposition d'unités de récupération GROUPE, demande la liste des transactions en attente de validation, la liste envoyée comprend toutes les transactions en attente de validation avec une disposition d'unités de récupération GROUPE qui existent par le biais du groupe de partage de files d'attente. Cette liste ne dépend pas du gestionnaire de files d'attente sur lequel ces transactions en attente de validation ont démarré. Un gestionnaire de files d'attente traitant une telle demande compile la liste en communiquant avec tous les autres gestionnaires de files d'attente actifs dans le groupe de partage de files d'attente à l'aide de la commande SYSTEM.QSG.UR.RESOLUTION.QUEUE. Le gestionnaire de files d'attente lit ensuite les journaux des gestionnaires inactifs à partir de leur dernier point de contrôle afin d'identifier toute transaction en attente de validation supplémentaire qu'ils auraient signalée s'ils avaient été actifs.

Lorsqu'un coordinateur de transactions demande la résolution d'une transaction en attente de validation, le gestionnaire de files d'attente auquel elle est connectée identifie si la transaction commence d'elle-même et, le cas échéant, la résout de la même façon que des transactions avec une disposition d'unités de récupération de QMGR. Si la transaction a commencé sur un autre gestionnaire de files d'attente actif dans le groupe de partage de files d'attente, une demande de fin de la résolution est acheminée vers le gestionnaire de files d'attente à l'aide de la commande SYSTEM.QSG.UR.RESOLUTION.QUEUE. Si la transaction a commencé sur un gestionnaire de files d'attente inactif dans le groupe de partage de files d'attente, tout travail de file d'attente partagée est résolu immédiatement et une demande de résolution de tout travail de file d'attente privée est placée sur la commande SYSTEM.QSG.UR.RESOLUTION.QUEUE. Le gestionnaire de files d'attente inactif traite cette demande au démarrage, avant d'accepter un nouveau travail. Dans ce cas, les journaux originaux du gestionnaire de files d'attente reflètent encore que l'unité de récupération est en attente de validation jusqu'au redémarrage et au traitement de la requête.

### **Récupération d'une file d'attente partagée**

Cette rubrique explique la récupération et la résilience IBM MQ de divers composants dans l'environnement de groupe de partage de files d'attente.

- [«Récupération transactionnelle»](#), à la page 282
- [«Reprise homologue»](#), à la page 282
- [«Définitions de file d'attente partagée»](#), à la page 283

- [«Journalisation», à la page 283](#)
- [«Echec de l'unité de couplage et des structures», à la page 283](#)
- [«Scénarios d'échec de la structure», à la page 284](#)
- [«Résilience en cas d'échec de connectivité d'une unité de couplage», à la page 285](#)
- [«Gestion de la résilience en cas d'échec de la connectivité d'une unité de couplage», à la page 286](#)
- [«Comportement fonctionnel», à la page 288](#)

## Récupération transactionnelle

Lorsqu'une application émet un appel MQBACK ou s'arrête de façon anormale (par exemple, en raison d'une annulation CICS EXEC ou d'un abandon IMS), les informations de niveau d'unité stockées dans le gestionnaire de files d'attente garantissent l'annulation de l'unité d'oeuvre en cours. Les opérations MQPUT et MQGET au sein d'un point de synchronisation sur les files d'attente partagées sont annulées de la même façon que les mises à jour des files d'attente non partagées.

## Reprise homologue

En cas d'échec d'un gestionnaire de files d'attente, celui-ci se déconnecte de façon anormale des structures d'unité de couplage auxquelles il est connecté. Si la connexion entre l'instance z/OS et l'unité de couplage échoue (par exemple, erreur de liaison physique ou interruption de l'unité de couplage ou d'une partition), cet incident est également détecté en tant qu'interruption anormale de la connexion entre le gestionnaire de files d'attente et les structures d'unité de couplage concernées. Les autres gestionnaires de files d'attente dans le même groupe de partage de files d'attente qui restent connectés à cette structure détectent la déconnexion anormale et toute tentative d'initiation d'une *reprise homologue* pour le gestionnaire de files d'attente qui a échoué sur cette structure. Seul un de ces gestionnaires de files d'attente initie une reprise homologue avec succès et tous les autres gestionnaires de files d'attente coopèrent pour la récupération d'unités d'oeuvre détenues par le gestionnaire de files d'attente qui a échoué.

Si un gestionnaire de files d'attente échoue alors qu'un homologue n'est connecté à une structure, la reprise est réalisée lorsqu'un autre gestionnaire de files d'attente se connecte à cette structure ou lorsque le gestionnaire qui a échoué redémarre.

La reprise homologue, souvent appelée PLR (Peer Level Recovery), est réalisée sur une structure en fonction d'une base de structure et il est possible pour un seul gestionnaire de files d'attente de participer à la récupération de plusieurs structures en même temps. Toutefois, l'ensemble d'homologues coopérant à la récupération de diverses structures peut varier en fonction des gestionnaires de files d'attente qui étaient connectés aux différentes structures au moment de l'échec.

Lorsque le gestionnaire de files d'attente qui a échoué redémarre, il se reconnecte aux structures auxquelles il était connecté au moment de l'échec et récupère toutes les unités d'oeuvre non résolues restantes qui n'ont pas été récupérées par la reprise homologue.

La reprise homologue s'effectue en plusieurs phases. Pendant la première phase, les unités d'oeuvre qui avaient progressé au-delà de la phase en cours sont récupérées ; cela peut impliquer de valider des messages pour les unités d'oeuvre qui sont en cours de validation et les messages de verrouillage pour les unités d'oeuvre en attente de validation. Pendant la seconde phase, les files d'attente qui avaient des unités d'exécution actives contre elles dans le gestionnaire de files d'attente qui a échoué sont vérifiées, les messages refusés relatifs aux unités d'oeuvre en cours sont annulés et les informations sur les indicateurs actifs des files d'attente partagées dans le gestionnaire qui a échoué sont réinitialisées. Cela signifie qu'IBM MQ réinitialise tous les indicateurs pour lesquels le gestionnaire de files d'attente avait une file d'attente partagée ouverte pour une insertion exclusive, permettant aux autres gestionnaires de files d'attente d'ouvrir la file d'attente pour des insertions.

## Définitions de file d'attente partagée

Les objets de file d'attente qui représentent les attributs d'une file d'attente partagée se trouvent dans le référentiel Db2 partagé, utilisé par le groupe de partage de files d'attente. Vérifiez que les procédures adéquates sont en place pour la sauvegarde et la récupération des tables Db2 qui contiennent les objets IBM MQ. Vous pouvez également utiliser l'utilitaire IBM MQ CSQUTIL pour créer des commandes MQSC à réexécuter dans un gestionnaire de files d'attente pour redéfinir les objets IBM MQ, y compris les définitions de file d'attente et de groupe stockées dans Db2.

## Journalisation

Les groupes de partage de files d'attente peuvent prendre en charge les messages persistants car les messages présents sur les files d'attente partagées peuvent être consignés dans les journaux du gestionnaire de files d'attente.

## Echec de l'unité de couplage et des structures

Deux types d'échec peuvent être signalés pour une structure d'unité de couplage : l'échec de la structure et la perte de connectivité. Les services Sysplex pour le partage de données (XES) informent IBM MQ de l'échec d'une structure d'unité de couplage ou d'une unité de couplage avec un événement d'échec de structure. Si XES crée un événement de perte de connectivité, cela n'indique pas nécessairement qu'il y a un problème avec la structure ; cela peut signifier qu'aucune connexion n'est disponible pour communiquer avec la structure. Il est possible que tous les gestionnaires de files d'attente ne reçoivent pas un événement de perte de connectivité pour la structure ; cela dépend de la configuration des connexions à l'unité de couplage. Un événement de perte de connectivité peut également être reçu en raison de commandes de l'opérateur, par exemple VARY PATH OFFLINE ou CONFIG CHP OFFLINE.

Les structures d'unités de couplage utilisées par IBM MQ peuvent être configurées pour utiliser le duplexage géré par le système. Cela signifie qu'en cas d'échec unique, le traitement de reprise géré par le système masque l'échec d'une structure ou la perte de connectivité et le gestionnaire de files d'attente n'est pas informé de l'échec. En cas d'échec des deux instances d'une structure duplexée ou d'une connexion, le gestionnaire de files d'attente reçoit l'événement approprié et le gère de la même façon qu'un événement d'échec pour une structure simplex. Les informations sur la façon dont le gestionnaire de files d'attente gère les événements sont décrites dans les [Scénarios](#).

Dans le cas peu probable d'un échec d'une unité de couplage ou d'une structure, tout message non persistant stocké dans les structures d'application concernées sont perdus. Vous pouvez récupérer les messages persistants à l'aide de la commande RECOVER CFSTRUCT. Si une structure d'application pouvant être récupérée échoue, toute activité d'application sur cette structure est évitée jusqu'à ce que la structure soit récupérée.

Pour vous assurer de pouvoir récupérer une structure d'unité de couplage dans un délai raisonnable, réalisez des sauvegardes fréquentes à l'aide de la commande BACKUP CFSTRUCT. Vous pouvez choisir d'effectuer des sauvegardes sur tous les gestionnaires de files d'attente dans le groupe de partage de files d'attente ou de dédier un gestionnaire de files d'attente qui effectuera toutes les sauvegardes. Automatisez le processus de sauvegarde pour veiller à ce qu'elles soient effectuées de façon régulière.

Chaque sauvegarde est écrite sur le jeu de données de journal actif du gestionnaire de files d'attente effectuant la sauvegarde. Le référentiel Db2 de file d'attente partagée enregistre le nom de la structure d'unité de couplage à sauvegarder, le nom du gestionnaire de files d'attente qui effectue la sauvegarde et la plage d'adresses relatives en octets de la sauvegarde dans le journal du gestionnaire de files d'attente et l'heure de la sauvegarde.

La structure d'administration contient des informations sur les unités d'oeuvre incomplètes des files d'attente partagées au moment de l'échec d'une structure d'application ; la structure d'administration doit donc être disponible lors du traitement RECOVER CFSTRUCT. Si la structure d'administration échoue, tous les gestionnaires de files d'attente dans le groupe de partage de files d'attente doivent avoir généré

à nouveau leurs entrées de structure d'administration avant de pouvoir publier la commande RECOVER CFSTRUCT.

Les gestionnaires de files d'attente régénèrent les entrées de leur structure d'administration automatiquement et sans s'arrêter. Si un gestionnaire de files d'attente ne s'exécute pas au moment de l'échec, les entrées de sa structure d'administration peuvent être régénérées par un autre gestionnaire de files d'attente qui se trouve dans le groupe de partage de files d'attente et dont le niveau est identique ou ultérieur.

Pour récupérer une structure d'application, publiez une commande RECOVER CFSTRUCT pour le gestionnaire de files d'attente qui réalisera la récupération. Vous pouvez récupérer une seule structure d'unité de couplage ou plusieurs structures d'unité de couplage simultanément. Vous pouvez procéder à la récupération avec n'importe quel gestionnaire de files d'attente se trouvant dans le groupe de partage de files d'attente ; il ne doit pas forcément s'agir de celui qui a effectué la sauvegarde ou de celui qui a été précédemment connecté à la structure qui a échoué.

La commande RECOVER CFSTRUCT utilise la sauvegarde située dans les informations du référentiel Db2 (Db2 doit donc être disponible sur le gestionnaire de files d'attente sur lequel la récupération a lieu) et récupère cela au moment de l'échec.

La commande RECOVER CFSTRUCT effectue cette opération en appliquant les enregistrements de journal de chaque gestionnaire de files d'attente dans le groupe de partage de files d'attente qui a réalisé un MQPUT ou un MQGET entre le début de la sauvegarde et le moment de l'échec, à n'importe quel gestionnaire de files d'attente partagée qui mappe à la structure d'unité de couplage. La fusion résultant des enregistrements nécessite la lecture d'un nombre considérable de données de journal car toutes les données de journal écrites en participant aux gestionnaires de files d'attente depuis la sauvegarde sont lues. Il vous est vivement conseillé d'effectuer des sauvegardes fréquentes (par exemple, toutes les heures), notamment s'il y a de grands messages au sein de la sauvegarde.

## Scénarios d'échec de la structure

### Scénarios

Si un échec est signalé pour une structure d'unité de couplage, l'action prise par les gestionnaires de files d'attente connectés dépend de ce qui suit :

- Le type d'échec signalé par le composant XES de z/OS à IBM MQ.
- Le type de structure (application ou administration)
- Le niveau du gestionnaire de files d'attente
- Le CFLEVEL de l'objet IBM MQ CFSTRUCT (2, 3, 4 ou 5. Il ne s'agit pas du CFLEVEL du microcode CFCC)
- L'attribut RECAUTO d'un objet IBM MQ CFSTRUCT au niveau CFLEVEL(5)

Les scénarios suivants décrivent ce qui se produit lorsqu'un échec est signalé pour la structure d'administration :

- Si un événement d'échec de structure est reçu pour la structure d'administration, la structure est réaffectée et régénérée automatiquement sans que le gestionnaire de files d'attente ne s'arrête. Une nouvelle instance de la structure est affectée par XES lorsqu'un gestionnaire de files d'attente tente de s'y connecter.

Lorsque le gestionnaire de files d'attente se connecte à une nouvelle instance de la structure, le gestionnaire de files d'attente écrit les entrées pour lui-même dans la structure. Ce traitement est effectué par le gestionnaire de files d'attente et ne fait pas partie du traitement de régénération de XES.

Si un gestionnaire de files d'attente n'était pas en cours d'exécution au moment de l'échec ou s'arrête avant la fin de la récupération de sa partie de la structure d'administration, ses entrées de structure d'administration peuvent être régénérées par un autre gestionnaire de files d'attente qui se trouve dans le groupe de partage de files d'attente.

Les entrées de la structure d'administration d'un gestionnaire de files d'attente ne peuvent être régénérées que par un autre gestionnaire de files d'attente dont le niveau d'exécution est identique ou ultérieur. Si les entrées de la structure d'administration d'un gestionnaire de files d'attente ne peuvent pas être régénérées par un autre gestionnaire de files d'attente qui se trouve dans le groupe de partage de files d'attente, redémarrez le gestionnaire de files d'attente pour qu'il puisse effectuer la régénération de sa partie de la structure.

Certaines actions sont interrompues jusqu'à ce que les entrées de la structure d'administration pour tous les gestionnaires de files d'attente aient été régénérées. Elles incluent les actions suivantes :

- Ouverture et fermeture des files d'attente.
- Validation ou annulation des unités de récupération.
- Applications numérotées se connectant ou se déconnectant du gestionnaire de files d'attente.
- Sauvegarde ou récupération d'une structure d'application.

Toute application numérotée déjà connectée au gestionnaire de files d'attente peut continuer le traitement. Toute application numérotée essayant de se connecter avec les paramètres MQCNO\_SERIALIZE\_CONN\_TAG\_QSG ou MQCNO\_RESTRICT\_CONN\_TAG\_QSG reçoit le code de retour MQRC\_CONN\_TAG\_NOT\_USABLE.

Lorsque les entrées de la structure d'administration pour le gestionnaire de files d'attente ont été régénérées, les actions suspendues reprennent.

Les scénarios suivants décrivent ce qui se produit lorsqu'un échec est signalé pour une structure d'administration :

- Si un événement d'échec de structure est reçu pour une structure d'application et que le CFLEVEL est 1 ou 2, le gestionnaire de files d'attente s'arrête. Redémarrez le gestionnaire de files d'attente. Le premier gestionnaire de files d'attente qui essaie de se connecter à nouveau à la structure entraîne l'affectation par XES d'une nouvelle instance de la structure.
- Si un événement d'échec de structure est reçu pour une structure d'application et que le CFLEVEL est 3, 4 ou 5, les gestionnaires de files d'attente connectés à la structure continuent de s'exécuter. Les applications qui n'utilisent pas les files d'attente dans la structure qui a échoué peuvent poursuivre un traitement normal.

Toutefois, les applications qui tentent des opérations sur les files d'attente de la structure qui a échoué reçoivent une erreur MQRC\_CF\_STRUC\_FAILED jusqu'à ce que la structure ait été régénérée correctement, auquel cas l'application peut à nouveau ouvrir les files d'attente.

La régénération de la structure est démarrée automatiquement pour les structures d'application CFLEVEL(5) définies avec RECAUTO(YES). Dans les autres cas, la structure est régénérée lorsque la commande RECOVER CFSTRUCT est émise.

## **Résilience en cas d'échec de connectivité d'une unité de couplage**

### **Qu'est ce que la résilience en cas d'échec de connectivité d'une unité de couplage ?**

La résilience en cas d'échec de connectivité d'une unité de couplage se définit comme la capacité des gestionnaires de files d'attente d'un groupe de partage de fichiers à tolérer cette perte de connectivité vers une structure d'unité de couplage sans interruption de la connexion. Cette fonction tente également de régénérer la structure dans une autre unité de couplage, mais avec une meilleure connectivité afin de rétablir dans les plus brefs délais l'accès aux files d'attente partagées.

### **Qu'est-ce qu'une perte partielle de connectivité ?**

IBM MQ définit une perte partielle de connectivité comme une situation dans laquelle un ou plusieurs systèmes du sysplex perdent la connectivité à l'unité de couplage à laquelle la structure à laquelle le système accède est allouée, alors que la connectivité entre un système du sysplex au moins et cette unité de couplage est conservée.

## Qu'est-ce qu'une perte totale de connectivité ?

IBM MQ définit une perte totale de connectivité comme une situation dans laquelle aucune connectivité n'existe entre les systèmes et l'unité de couplage et la structure allouée dans celle-ci.

## Pourquoi activer cette fonction ?

La résilience en cas d'échec de la connectivité d'une unité de couplage permet d'améliorer la disponibilité d'IBM MQ, en permettant aux files d'attente non partagées de rester disponibles en cas de perte de la connectivité entre un gestionnaire de fichiers et une ou plusieurs structures d'unité de couplage. De plus, les gestionnaires de files d'attente qui perdent la connectivité vers une structure d'unité de couplage tentent automatiquement de régénérer cette structure dans une autre unité de couplage disponible, ce qui permet d'améliorer la disponibilité des files d'attente partagées dans le groupe de partage de files d'attente.

## Considérations liées à l'activation de cette fonction

Un gestionnaire de files d'attente qui tolère la perte de connectivité à des structures d'unité de couplage sans s'arrêter risque de ne pas pouvoir se reconnecter à une telle structure pendant un certain temps si aucune autre unité de couplage n'est disponible. Les files d'attente partagées définies sur une structure ayant souffert d'une perte de connectivité restent indisponibles jusqu'à la restauration de la connectivité à la structure. Dans une telle situation, les applications se connectant à des membres du groupe de partage de files d'attente pour y exécuter des tâches de la file d'attente partagée risquent de se heurter à une indisponibilité des files d'attente auxquelles elles doivent accéder. Pour éviter ce cas de figure, nous vous recommandons de configurer les gestionnaires de files d'attente de telle sorte qu'ils s'arrêtent en cas de perte de la connectivité à une structure d'unité de couplage. Cet arrêt force les applications à se connecter à un autre membre du groupe de partage de files d'attente connecté aux structures d'unité de couplage dans lesquelles les files d'attente partagées requises par l'application sont définies.

## Gestion de la résilience en cas d'échec de la connectivité d'une unité de couplage

### Comment activer cette fonction ?

Exécutez la procédure suivante pour activer la résilience en cas d'échec de la connectivité de l'unité de couplage.

1. Vérifiez que le fichier CDS CFRM a été formaté pour prendre en charge la régénération gérée par le système. Les gestionnaires de files d'attente peuvent alors initier cette régénération afin de recréer une structure dans une unité de couplage disponible. Utilisez la commande **DISPLAY XCF, COUPLE, TYPE=CFRM** pour déterminer le format du fichier CDS CFRM. Pour que la régénération gérée par le système soit prise en charge, le fichier CDS CFRM doit être formaté via la commande :

```
"ITEM NAME(SMREBLD) NUMBER(1) "
```

Pour plus d'informations sur le formatage d'un fichier de couple CFRM, voir la documentation [z/OS MVS Setting Up a Sysplex](#) .

2. Vérifiez qu'une autre unité de couplage est disponible et qu'elle figure dans la liste des préférences CFRM pour toutes les structures d'unité de couplage IBM MQ. Les gestionnaires de files d'attente pourront ainsi régénérer les structures dans une autre unité de couplage disponible afin de restaurer l'accès à celles-ci le plus rapidement possible.

Les structures IBM MQ doivent être définies avec ENFORCEORDER(NO) dans la règle CFRM afin que XCF puisse choisir la structure d'unité de couplage optimale dans la configuration si IBM MQ doit réaffecter la structure.

Pour plus d'informations sur les listes de préférences de structure, voir la documentation [z/OS MVS Setting Up a Sysplex](#) .

3. Affectez le niveau CFLEVEL(5) à toutes les structures d'unité de couplage de l'application devant tolérer une perte de connectivité. Il s'agit du niveau minimal pouvant tolérer une telle perte.

4. Déterminez les valeurs requises pour les attributs **QMGR CFCONLOS** et **CFSTRUCT CFCONLOS** et modifiez-les en conséquence. L'attribut **QMGR CFCONLOS** permet de contrôler la tolérance de la perte de connectivité à la structure d'administration et l'attribut **CFSTRUCT CFCONLOS** permet de contrôler la tolérance de la perte de connectivité par chaque structure d'unité de couplage de l'application. Si les valeurs associées par défaut à ces attributs sont conservées, le gestionnaire de files d'attente s'arrête en cas de perte de connectivité à l'une des structures de l'unité de couplage.
5. Déterminez les valeurs requises pour l'attribut **CFSTRUCT RECAUTO** pour chaque structure d'unité de couplage de l'application et modifiez-les en conséquence. Cet attribut permet de contrôler si les structures d'unité de couplage doivent être automatiquement récupérées à l'aide des données consignées en cas de perte de connectivité. Si la valeur associée par défaut à cet attribut est conservée, aucune récupération automatique ne sera effectuée pour les structures d'application dans un tel cas.

### Scénario 1 - Perte de connectivité à la structure d'administration

Les gestionnaires de file d'attente peuvent tolérer une perte de connectivité à la structure d'administration sans devoir s'arrêter si tous les gestionnaires de files d'attente du groupe de partage de files d'attente sont au niveau de la IBM WebSphere MQ 7.1 ou à un niveau ultérieur. Si des gestionnaires de files d'attente se trouvent à un niveau antérieur à la IBM WebSphere MQ 7.1 dans le groupe de partage de files d'attente, tous les gestionnaires de files d'attente dans le groupe s'arrêtent de façon anormale avec le code anomalie 00C510AB en cas de perte de connectivité à la structure d'administration.

En cas de perte de la connectivité vers la structure d'administration par un gestionnaire de files d'attente configuré pour la tolérer, tous les membres du groupe de partage de files d'attente se déconnectent de la structure d'administration. Tous les gestionnaires de files d'attente du groupe tentent alors de se reconnecter à la structure d'administration, qui est réaffectée à tous les systèmes du sysplex avec la meilleure connectivité possible, tandis que les données de structure d'administration sont régénérées.

**Remarque :** L'unité de couplage dont la connectivité à tous les systèmes est la meilleure n'est pas nécessairement celle dont les gestionnaires de files d'attente sont actifs.

Si un gestionnaire de files d'attente ne parvient pas à se reconnecter à la structure d'administration, par exemple parce qu'aucune unité de couplage de la liste des préférences CFRM de la structure d'administration n'est disponible, certaines opérations liées à la file d'attente partagée resteront indisponibles jusqu'au rétablissement de la connexion et à la régénération des données de la structure d'administration. La reconnexion s'effectue automatiquement lorsqu'une unité de couplage appropriée devient disponible sur le système.

Aucun échec de la connexion à la structure d'administration lors du démarrage du gestionnaire de files d'attente en raison de l'absence de connectivité à l'unité de couplage ou de l'absence d'une unité de couplage appropriée pour allouer la structure n'est toléré. Tous les gestionnaires de files d'attente du groupe tentent alors de se reconnecter à la structure d'administration, qui est réaffectée à une autre unité de couplage disponible, tandis que les données de structure d'administration sont régénérées.

### Scénario 2- Perte de connectivité à la structure d'application

Une perte de connectivité à les structures d'application de niveau **CFLEVEL (5)** ou d'un niveau supérieur peut être tolérée sans entraîner l'arrêt du gestionnaire de files d'attente. Les gestionnaires de files d'attente connectés à des structures d'application de niveau **CFLEVEL (4)** ou inférieur, ou des structures de niveau **CFLEVEL (5)** qui n'ont pas été configurées pour tolérer une perte de connectivité, se terminent de manière anormale avec le code anomalie 00C510AB lorsque la connectivité à la structure est perdue.

En cas de perte de la connectivité à une structure d'application configurée pour la tolérer, tous les gestionnaires de files d'attente perdant la connectivité se déconnectent. Selon qu'il s'agit d'une perte de connectivité partielle ou totale, le comportement futur du gestionnaire de files d'attente varie.

#### Perte partielle de connectivité à une structure d'application

Si la perte de connectivité s'avère être partielle, les gestionnaires de files d'attente concernés tentent d'initier une régénération gérée par le système afin de déplacer la structure vers une autre unité de couplage dont la connectivité est meilleure. Si cette régénération aboutit, les messages persistants et non persistants de la structure sont copiés vers l'autre unité de couplage et l'accès aux files d'attente de la structure est restauré. Les gestionnaires de files d'attente qui n'ont pas perdu leur connectivité restent connectés à la structure ; toutefois, les opérations accédant à la structure sont différées au cours du processus de régénération géré par le système.

S'il est impossible de régénérer une structure d'application sur une unité de couplage bénéficiant d'une meilleure connectivité ou si la connectivité entre certains gestionnaires de files d'attente et la structure régénérée sur une autre unité de couplage n'est toujours pas établie, les files d'attente définies sur la structure restent indisponibles sur ces gestionnaires de files d'attente jusqu'à la restauration de la connectivité à l'unité de couplage. Les gestionnaires de files d'attente se reconnectent automatiquement à la structure lorsque celle-ci devient disponible et l'accès aux files d'attente partagées définies sur cette structure est alors rétabli.

### **Perte totale de connectivité à une structure d'application**

En cas de perte de la connectivité entre tous les systèmes du sysplex et l'unité de couplage dans laquelle la structure d'application est affectée, z/OS désaffecte la structure de l'unité de couplage chaque fois qu'une tentative de reconnexion à la structure est effectuée. Le gestionnaire de files d'attente peut tenter de se reconnecter à la structure pour différentes raisons, par exemple lorsqu'une application tente d'ouvrir une file d'attente partagée ou lorsque le système envoie une notification indiquant que de nouvelles ressources de l'unité de couplage sont peut-être disponibles. Il est donc probable que tous les messages non persistants se trouvant dans la structure concernée soient perdus en cas de perte totale de la connectivité à une structure d'application.

Si elles ont été définies avec **RECAUTO(YES)**, les structures d'application récupérables sont automatiquement récupérées après une perte totale de connectivité. La récupération démarre presque immédiatement si une autre unité de couplage à laquelle la structure peut être affectée est disponible ou dès qu'une telle unité de couplage devient disponible. Si une structure n'a pas été définie avec **RECAUTO(YES)**, la reprise peut être démarrée à l'aide de la commande **RECOVER CFSTRUCT**. Tous les messages persistants de la structure sont alors récupérés, tandis que les messages non persistants sont perdus. Cette opération impliquant la lecture du journal du gestionnaire de files d'attente, son exécution peut prendre un certain temps. Nous vous recommandons donc d'effectuer des sauvegardes régulières de la structure afin de réduire le temps d'attente préalable à la restauration de l'accès aux files d'attente partagées.

Les gestionnaires de files d'attente tentent de se reconnecter aux structures d'application non récupérables dès qu'une application tente d'ouvrir un file d'attente partagée définie sur la structure ou dès que le système envoie une notification indiquant que de nouvelles ressources de l'unité de couplage sont devenues disponibles. Si une unité de couplage appropriée à laquelle une structure peut être affectée devient disponible, une nouvelle structure est affectée et l'accès aux files d'attente partagées définies sur celle-ci est restauré. Les messages persistants ne pouvant être placés dans les files d'attente définies sur les structures non récupérables, tous les messages présents dans les files d'attente sont perdus.

## **Comportement fonctionnel**

Si un gestionnaire de files d'attente de la IBM WebSphere MQ 7.1 ou ultérieure, configuré pour tolérer une perte de connectivité à une structure d'unité de couplage perd la connexion, les membres du groupe de partage de files d'attente tentent automatiquement de procéder à une récupération en cas d'échec et de se reconnecter à la structure. Si une unité de couplage dont la connectivité est meilleure est disponible, il sera peut-être nécessaire de réaffecter cette structure dans une unité de ce type. Toutefois, un opérateur devra probablement intervenir afin de procéder à la récupération.

En règle générale, il devra exécuter les actions suivantes :

1. Résoudre la cause de l'échec ayant entraîné la perte de connectivité.

2. Vérifier qu'une unité de couplage dans laquelle les structures IBM MQ peuvent être affectées est disponible sur tous les systèmes du sysplex.

Les structures ayant été automatiquement réaffectées dans une autre unité de couplage après la perte de connectivité peuvent être déplacées vers l'unité de couplage bénéficiant de la connectivité optimale vers tous les gestionnaires de files d'attente du groupe de partage de files d'attente. Si cette opération est nécessaire, exécutez la commande de régénération gérée par le système **SETXCF START, REBUILD**, conformément aux instructions fournies dans z/OS MVS System Commands Reference.

En cas de perte partielle de connectivité à une structure d'application, les gestionnaires de files d'attente concernés tentent d'initier une régénération gérée par le système. Ce processus affecte la structure à une autre unité de couplage uniquement si la connectivité entre celle-ci et tous les gestionnaires de files d'attente actuellement connectés existe. Il est donc possible qu'en cas de perte de la connectivité entre la majorité des gestionnaires de files d'attente d'un groupe de partage de files d'attente et une structure d'application, ces gestionnaires ne parviennent pas à régénérer la structure dans une autre unité de couplage du fait qu'ils sont toujours connectés à la structure d'origine. Dans ce cas de figure, il est possible d'arrêter les gestionnaires de files d'attente toujours connectés à la structure d'origine afin de permettre la régénération de la structure, ou d'émettre la commande **RESET CFSTRUCT ACTION(FAIL)** pour entraîner l'échec de la structure. Pour procéder à une récupération sur les structures applicables, vous pouvez émettre la commande **RECOVER CFSTRUCT**.

**Remarque :** En cas d'échec et de récupération de la structure, tous les messages non persistants se trouvant sur la structure sont perdus.

z/OS

## Concepts de sécurité dans IBM MQ for z/OS

Cette rubrique explique l'importance de la sécurité pour IBM MQ et les conséquences liées à l'absence de paramètres de sécurité adéquats sur votre système.

### Raisons pour lesquelles vous devez protéger les ressources IBM MQ

IBM MQ gère le transfert d'informations qui peuvent s'avérer précieuses. L'application de la sécurité garantit que les ressources qu'IBM MQ possède et gère sont protégées contre les accès non autorisés. De tels accès peuvent entraîner une perte ou une divulgation des informations.

Vous devez vérifier qu'aucune des ressources suivantes n'est ouverte ou modifiée par un utilisateur ou un processus non autorisé :

- Les connexions à IBM MQ.
- Les objets IBM MQ tels que les files d'attente, les processus et les listes de noms.
- Les liaisons de transmission IBM MQ, c'est-à-dire les canaux IBM MQ.
- Les commandes de contrôle du système IBM MQ.
- Messages IBM MQ
- Les informations contextuelles associées aux messages.

Afin de fournir la sécurité adéquate, IBM MQ fait appel à la fonction d'autorisation système (SAF) z/OS pour acheminer les demandes d'autorisation vers un gestionnaire de sécurité externe (ESM), tel que Security Server (anciennement appelé RACF). IBM MQ n'effectue aucune vérification lui-même. Lorsque la mise en files d'attente réparties ou les clients sont utilisés, il se peut que vous ayez besoin d'une sécurité accrue, pour laquelle IBM MQ fournit des enregistrements d'authentification de canal, des exits de canal, l'attribut de canal MCAUSER et le protocole TLS.

La décision d'autoriser l'accès à un objet revient au gestionnaire de sécurité externe (ESM) et IBM MQ suit cette décision. Si le gestionnaire ESM ne parvient pas à prendre une décision, IBM MQ empêche l'accès à l'objet.

## Conséquences de la non-protection des ressources IBM MQ ?

Si vous ne prenez aucune mesure de sécurité, *tous* les utilisateurs peuvent accéder à *toutes* les ressources et les modifier. Cela comprend également les utilisateurs présents sur les systèmes distants faisant appel à la mise en files d'attente réparties ou aux clients, dans lesquels les contrôles de sécurité de connexion sont moins stricts que sous z/OS.

Pour activer le contrôle de la sécurité, procédez comme suit :

- Installez un gestionnaire de sécurité externe (par exemple, Security Server) et activez-le.
- Définissez la classe MQADMIN si vous avez recours à un gestionnaire de sécurité externe autre que Security Server.
- Activez la classe MQADMIN.

Vous devez décider si l'utilisation de noms de ressources faisant la distinction entre les minuscules et les majuscules présente un quelconque avantage pour votre entreprise. Si vous choisissez d'utiliser des noms faisant la distinction entre les minuscules et les majuscules dans vos profils ESM, vous devez définir et activer la classe MXADMIN.

## Chiffrement de fichiers z/OS

Data Set Encryption (DSE) permet de chiffrer des fichiers z/OS, de sorte que les données qu'ils contiennent ne puissent être affichées et modifiées que par les ID utilisateur disposant des droits spécifiques. Cela permet le chiffrement des données statiques dans le système de fichiers et empêche toute divulgation accidentelle des informations sensibles aux utilisateurs ayant un besoin professionnel légitime et disposant de droits de gestion des fichiers.

**CD** **V 9.2.0** Avant la IBM MQ for z/OS 9.1.4, IBM MQ for z/OS ne prend pas en charge l'utilisation de DSE avec les journaux actifs, les ensembles de pages et les fichiers de messages partagés (SMDS) qui fournissent les mécanismes de persistance principaux des messages IBM MQ.

Au lieu de cela, Advanced Message Security fournit une solution de chiffrement de bout en bout pour la messagerie IBM MQ, qui englobe l'intégralité du réseau IBM MQ, le chiffrement des données à la volée, des données statiques et même à l'intérieur des processus IBM MQ de l'environnement d'exécution.

Les autres fichiers VSAM et séquentiels utilisés dans un sous-système IBM MQ peuvent être chiffrés à l'aide de DSE. Exemple :

- Le fichier d'amorce (BSDS)
- Les fichiers séquentiels contenant les commandes de configuration système (MQSC) lues au démarrage à l'aide de CSQINP<sub>x</sub> DDNAMEs
- Les journaux d'archivage IBM MQ, souvent utilisés pour l'archivage à long terme des données de journal IBM MQ à des fins d'audit.

Vous pouvez les chiffrer à l'aide de DSE en allouant une classe de données définie avec un intitulé de clé de fichier. Pour plus d'informations, voir [Planning your log archive storage](#).

**V 9.2.0** Depuis IBM MQ for z/OS 9.1.4, IBM MQ for z/OS prend en charge l'utilisation de DSE avec les journaux actifs et les ensembles de pages, en plus des éléments déjà pris en charge dans les éditions précédentes.

**V 9.2.0** IBM MQ for z/OS ne prend pas en charge l'utilisation de DSE pour les fichiers de messages partagés (SMDS).

**V 9.2.0** Voir la section [Confidentiality for data at rest on IBM MQ for z/OS with data set encryption](#) pour plus d'informations.

### Concepts associés

[Concepts de sécurité](#)

[Enregistrements d'authentification de canal](#)

[Droits d'utiliser les objets IBM MQ sur z/OS](#)

[Protocole de sécurité cryptographique TLS](#)

[Sources à partir desquelles vous pouvez émettre des commandes MQSC sous z/OS](#)

#### **Tâches associées**

[Configuration de la sécurité sous z/OS](#)

[Comparatif de la sécurité au niveau des liaisons et de la sécurité au niveau de l'application](#)

#### **Référence associée**

[Messages pour IBM MQ for z/OS](#)

## **Options et contrôles de sécurité dans IBM MQ for z/OS**

Vous pouvez spécifier si la sécurité doit être activée pour tout le sous-système IBM MQ et si vous voulez effectuer des vérifications de sécurité au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente. Vous pouvez également contrôler le nombre d'ID utilisateur vérifiés pour la sécurité des ressources d'API.

### **Sécurité du sous-système**

La sécurité du sous-système est un contrôle qui spécifie si des vérifications de sécurité sont effectuées pour l'intégralité du gestionnaire de files d'attente. Si vous ne demandez pas de vérification de la sécurité (par exemple, sur un système de test) ou si le niveau de sécurité sur toutes les ressources qui peuvent se connecter à IBM MQ vous convient (notamment les clients et les canaux), vous pouvez désactiver la vérification de sécurité pour le gestionnaire de files d'attente ou le groupe de partage de files d'attente afin ne pas vérifier la sécurité.

Il s'agit de la seule vérification qui peut désactiver la sécurité complètement et déterminer si d'autres vérifications de la sécurité doivent être réalisées. Autrement dit, si vous désactivez la vérification pour le gestionnaire de files d'attente ou le groupe de partage de files d'attente, aucune autre vérification d'IBM MQ n'est effectuée ; si vous la laissez activée, IBM MQ vérifie vos exigences de sécurité pour les autres ressources IBM MQ.

Vous pouvez également activer ou désactiver la sécurité pour des ensembles particuliers de ressources, tels que les commandes.

### **Vérification du niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente**

Vous pouvez implémenter la sécurité au niveau du gestionnaire de files d'attente ou au niveau du groupe de partage de files d'attente. Si vous implémentez la sécurité au niveau du groupe de partage de files d'attente, tous les gestionnaires de files d'attente dans le groupe partagent les mêmes profils. Cela signifie qu'il y a moins de profils à définir et à maintenir, facilitant ainsi la gestion de la sécurité. Cela facilite également l'ajout d'un nouveau gestionnaire de files d'attente au groupe de partage de files d'attente car il hérite des profils de sécurité existants.

Il est également possible d'implémenter une combinaison des deux si votre installation le requiert, par exemple lors de la migration ou si vous disposez d'un gestionnaire de files d'attente dans le groupe de partage de files d'attente qui requiert plusieurs niveaux de sécurité pour les autres gestionnaires de files d'attente dans le groupe.

#### **Sécurité au niveau du groupe de partage de files d'attente**

La vérification de la sécurité au niveau du groupe de partage de files d'attente est effectuée sur l'intégralité du groupe. Cela vous permet de simplifier l'administration de la sécurité dans la mesure où vous devez définir moins de profils de sécurité. L'autorisation d'un ID utilisateur à utiliser une ressource particulière est gérée au niveau du groupe de partage de files d'attente et est indépendante du gestionnaire de files d'attente qu'utilise l'ID utilisateur pour accéder à la ressource.

Supposons qu'une application de serveur fonctionnant sous ID SERVER souhaite accéder à la file d'attente SERVER.REQUEST et que vous vouliez exécuter une instance de SERVER sur chaque image z/OS dans le sysplex. Plutôt que d'autoriser SERVER à ouvrir SERVER.REQUEST sur chaque gestionnaire de files d'attente individuellement (sécurité au niveau du gestionnaire de files d'attente), vous pouvez autoriser l'accès uniquement au niveau du groupe de partage de files d'attente.

Vous pouvez utiliser les profils de sécurité au niveau du groupe de partage de files d'attente pour protéger tous les types de ressources, qu'elles soient locales ou partagées.

### **Sécurité au niveau du gestionnaire de files d'attente**

Vous pouvez utiliser les profils de sécurité au niveau des gestionnaires de files d'attente pour protéger tous les types de ressources, qu'elles soient en local ou partagées.

### **Combinaison des deux niveaux**

Vous pouvez utiliser une combinaison de la sécurité au niveau du gestionnaire de files d'attente et au niveau du groupe de partage de files d'attente.

Vous pouvez redéfinir les paramètres de sécurité au niveau du groupe de partage de files d'attente pour un gestionnaire de files d'attente particulier membre de ce groupe. Cela signifie que vous pouvez réaliser un niveau de vérifications de sécurité sur un gestionnaire de files d'attente individuel différent des vérifications réalisées sur les autres gestionnaires de files d'attente du groupe.

Pour plus d'informations, voir [Profils destinés au contrôle de la sécurité au niveau du groupe de partage de files d'attente ou du gestionnaire de files d'attente](#).

## **Contrôle du nombre d'ID utilisateur vérifiés**

RESLEVEL est le profil Serveur de sécurité qui contrôle le nombre d'ID utilisateur vérifiés pour la sécurité des ressources IBM MQ. Normalement, lorsqu'un utilisateur essaie d'accéder à une ressource IBM MQ, le serveur de sécurité vérifie l'ID utilisateur associé pour s'assurer que l'accès à cette ressource est autorisé. En définissant un profil RESLEVEL, vous pouvez contrôler si aucun, un ou, lorsque possible, deux ID utilisateur sont vérifiés.

Ces contrôles sont effectués sur une connexion, sur une base de connexion et s'étendent sur la durée de la connexion.

Il n'existe qu'un seul profil RESLEVEL pour chaque gestionnaire de files d'attente. Le contrôle est implémenté par l'accès d'un ID utilisateur à ce profil.

## **Classes en minuscules et majuscules ou en majuscules IBM MQ RACF**

Désormais, vous pouvez utiliser un support de profil RACF à casse mixte qui permet d'utiliser des noms de ressources en minuscules et majuscules et de définir des profils IBM MQ RACF pour les protéger.

Vous pouvez choisir de :

- Continuer à utiliser les classes IBM MQ RACF en majuscules uniquement comme dans les versions précédentes, ou
- Utiliser les nouvelles classes IBM MQ RACF en majuscules et minuscules.

Sans utiliser de profil en minuscules et majuscules RACF, vous pouvez toujours utiliser des noms de ressources en minuscules et en majuscules dans IBM MQ for z/OS, mais ces noms de ressource peuvent être protégés uniquement par des profils génériques RACF dans les classes IBM MQ en majuscules. Lorsque vous utilisez le support de profil IBM MQ RACF en minuscules et majuscules, vous pouvez fournir un meilleur niveau de protection en définissant des profils IBM MQ RACF dans des classes IBM MQ en minuscules et majuscules.

## Ressources que vous pouvez protéger dans IBM MQ for z/OS

Lors du démarrage d'un gestionnaire de files d'attente ou de l'exécution d'une commande de l'opérateur, IBM MQ for z/OS détermine les ressources à protéger.

Vous pouvez définir les contrôles de sécurité à effectuer sur chaque gestionnaire de files d'attente. Par exemple, vous pouvez implémenter plusieurs contrôles de sécurité sur un gestionnaire de files d'attente de production, et aucun sur le gestionnaire de files d'attente de test.

### Sécurité des connexions

Le contrôle de la sécurité liée aux connexions est exécuté lorsqu'un programme d'application tente de se connecter à un gestionnaire de files d'attente. Ce contrôle est obtenu en exécutant une demande MQCONN ou MQCONNX ou lorsque l'initiateur de canal ou l'adaptateur CICS ou IMS exécute une demande de connexion.

Si vous faites appel à la sécurité au niveau du gestionnaire de files d'attente, il vous est possible de désactiver le contrôle de la sécurité associé à un gestionnaire particulier. Dans ce cas, notez que tout utilisateur peut se connecter au gestionnaire de files d'attente.

Dans le cadre de l'adaptateur CICS, seul l'ID utilisateur d'espace adresse CICS est utilisé pour le contrôle de la sécurité des connexions, et non l'ID utilisateur du terminal CICS. Dans le cadre de l'adaptateur IMS, lorsque le contrôle IMS ou les régions dépendantes se connectent à IBM MQ, l'ID utilisateur d'espace adresse IMS est vérifié. Dans le cadre de l'initiateur de canal, l'ID utilisateur utilisé par l'espace adresse de l'initiateur de canal est vérifié.

Vous pouvez activer ou désactiver la vérification de la sécurité des connexions au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

### Sécurité de commande

Le contrôle de la sécurité de commande est effectué lorsqu'un utilisateur émet une commande MQSC depuis n'importe quelle source décrite dans [Emission de commandes](#). Vous pouvez effectuer un contrôle individuel sur la ressource spécifiée par la commande, tel que décrit dans [«Sécurité des ressources de commandes»](#), à la page 293.

Si vous désactivez le contrôle de commande, les utilisateurs ayant émis les commandes ne sont pas contrôlés afin de vérifier s'ils disposent ou non des droits appropriés.

Si les commandes MQSC sont saisies depuis une console, cette dernière doit posséder l'attribut de l'autorité SYS z/OS. Les commandes exécutées depuis les fichiers CSQINP1 ou CSQINP2, ou de manière interne par le gestionnaire de files d'attente, ne sont soumises à aucun contrôle de sécurité alors que celles exécutées depuis le fichier CSQINPX utilisent l'ID utilisateur d'espace adresse de l'initiateur de canal. Vous devez contrôler les permissions de mise à jour de ces fichiers par le biais de la protection normale des fichiers.

Vous pouvez activer ou désactiver la vérification de la sécurité des commandes au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

### Sécurité des ressources de commandes

Certaines commandes MQSC, permettant par exemple de définir une file d'attente locale, impliquent certaines manipulations de ressources IBM MQ. Lorsque la sécurité des ressources de commandes est active, chaque fois qu'une commande impliquant une ressource est exécutée, IBM MQ vérifie si l'utilisateur est autorisé à modifier la définition de cette ressource.

La sécurité des ressources de commandes permet de respecter les normes de dénomination. Par exemple, un administrateur de paie peut être habilité à supprimer et à définir uniquement les files d'attente dont le nom commence par "PAIE". Si la sécurité des ressources de commandes est inactive,

aucune vérification n'est effectuée sur la ressource manipulée par la commande. Ne confondez pas sécurité des ressources de commandes et sécurité de commande, ces deux concepts sont indépendants.

La désactivation de la sécurité des ressources de commandes n'a aucune incidence sur la vérification des ressources qui est effectuée pour d'autres types de traitement n'impliquant pas des commandes.

Vous pouvez activer ou désactiver la vérification de la sécurité des ressources de commandes au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

## Remarques sur la sécurité des canaux

### Sécurité des canaux

Lorsque vous faites appel aux canaux, les fonctions de sécurité varient en fonction du protocole de communication que vous sélectionnez. Si vous choisissez le protocole TCP, aucune fonction de sécurité n'est fournie avec le protocole de communication, mais vous avez la possibilité d'utiliser TLS. Si vous choisissez le protocole de communication évoluée de programme à programme (APPC), vous pouvez transmettre via le réseau les informations d'ID utilisateur de l'agent MCA source vers l'agent MCA cible à des fins de vérification.

Dans un cas comme dans l'autre, vous pouvez indiquer les ID utilisateur que vous souhaitez vérifier. Les choix disponibles dépendent également du protocole que vous utilisez, des éléments que vous avez spécifiés lors de la définition du canal et des paramètres RESLEVEL associés à l'initiateur de canal.

Pour plus d'informations sur les types de sécurité de canal disponibles, voir [Enregistrements d'authentification de canal](#) et [Présentation de l'exit de sécurité](#).

### Référence associée

«Sécurité des ressources d'API dans IBM MQ for z/OS», à la page 294

Les ressources sont vérifiées lorsqu'une application ouvre un objet avec un appel MQOPEN ou MQPUT1. L'accès nécessaire pour ouvrir un objet dépend des options d'ouverture spécifiées lorsque s'ouvre le gestionnaire de files d'attente.

### **Sécurité des ressources d'API dans IBM MQ for z/OS**

Les ressources sont vérifiées lorsqu'une application ouvre un objet avec un appel MQOPEN ou MQPUT1. L'accès nécessaire pour ouvrir un objet dépend des options d'ouverture spécifiées lorsque s'ouvre le gestionnaire de files d'attente.

La sécurité des ressources d'API se divise en les vérifications suivantes :

- [File d'attente](#)
- [Processus](#)
- [Liste de noms](#)
- [Utilisateur substitut](#)
- [Contexte](#)

Aucune vérification de sécurité n'est effectuée lors de l'ouverture de l'objet de gestionnaire de files d'attente ou de l'accès à des objets de classe de stockage.

### File d'attente

La vérification de la sécurité des files d'attente contrôle qui peut ouvrir telle file d'attente et quelles options peuvent être ouvertes. Par exemple, un utilisateur peut être autorisé à ouvrir une file d'attente appelée PAYROLL.INCREASE.SALARY pour parcourir les messages dans la file (à l'aide de l'option MQOO\_BROWSE) mais ne peut pas supprimer des messages de cette file d'attente (à l'aide de l'une des options MQOO\_INPUT\_\*). Si vous désactivez la vérification pour les files d'attente, n'importe quel utilisateur peut ouvrir une file d'attente avec une option d'ouverture valide (c'est-à-dire, toute option MQOO\_\* valide sur un appel MQOPEN ou MQPUT1).

Vous pouvez activer ou désactiver la vérification de la sécurité au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

## Processus

La vérification de la sécurité des processus est effectuée lorsqu'un utilisateur ouvre un objet de définition de processus. Si vous désactivez la vérification des processus, n'importe quel utilisateur peut ouvrir un processus.

Vous pouvez activer ou désactiver la vérification de la sécurité des processus au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

## Liste de noms

La vérification de la sécurité des listes de noms est effectuée lorsqu'un utilisateur ouvre une liste de noms. Si vous désactivez la vérification des listes de noms, n'importe quel utilisateur peut ouvrir une liste de noms.

Vous pouvez activer ou désactiver la vérification de la sécurité des listes de noms au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

## Utilisateur de remplacement

La sécurité d'utilisateur substitut contrôle si un ID utilisateur peut bénéficier des droits d'accès d'un autre ID utilisateur pour ouvrir un objet IBM MQ.

Exemple :

- Un programme serveur exécuté sous l'ID utilisateur PAYSERV récupère un message de demande d'une file d'attente inséré dans la file par l'ID utilisateur UTILISATEUR1.
- Lorsque le programme serveur obtient le message de demande, il traite la demande et insère la réponse dans la file d'attente de réponse spécifiée dans le message de demande.
- Plutôt que d'utiliser son propre ID utilisateur (PAYSERV) pour autoriser l'ouverture de la file d'attente de réponse, le serveur peut indiquer un autre ID utilisateur, dans ce cas UTILISATEUR1. Dans cet exemple, la sécurité d'utilisateur substitut contrôle si l'ID utilisateur PAYSERV a le droit d'indiquer l'ID utilisateur UTILISATEUR1 comme ID utilisateur substitut lors de l'ouverture de la file d'attente de réponse.

L'ID utilisateur substitut est indiqué dans la zone *AlternateUserId* du descripteur d'objet (MQOD).

Vous pouvez utiliser des ID d'utilisateur substituts sur n'importe quel objet IBM MQ, tels que des processus et des listes de noms. Cela n'affecte pas l'ID utilisateur utilisé par les autres gestionnaires de ressources, par exemple, pour la sécurité de CICS ou la sécurité des fichiers z/OS.

Si la sécurité d'utilisateur substitut n'est pas activée, n'importe quel utilisateur peut utiliser un autre ID utilisateur comme un ID utilisateur alternatif.

Vous pouvez activer ou désactiver la vérification de la sécurité d'utilisateur substitut au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

## Contexte

Le contexte est une information applicable à un message particulier et se trouvant dans le descripteur de message (MQMD) faisant partie du message. Les informations contextuelles comportent deux sections :

### La section d'identité

L'utilisateur de l'application qui insère en premier le message dans une file d'attente. Elle comprend les zones suivantes :

- *UserIdentifier*
- *AccountingToken*

- *ApplIdentityData*

### Section d'origine

L'application ayant inséré le message dans la file d'attente dans laquelle il est actuellement stocké. Elle comprend les zones suivantes :

- *PutApplType*
- *PutApplName*
- *PutDate*
- *PutTime*
- *ApplOriginData*

Les applications peuvent indiquer les données contextuelles lors d'un appel MQPUT ou MQPUT1. L'application peut générer les données, les données peuvent être transmises d'un autre message ou le gestionnaire de files d'attente peut générer les données par défaut. Par exemple, les programmes serveur peuvent utiliser les données contextuelles pour vérifier l'identité du demandeur : le message provient-il de l'application correcte ? Généralement, la zone *UserIdentifier* permet de déterminer l'ID utilisateur d'un utilisateur substitut.

Vous utilisez la sécurité du contexte pour contrôler si l'utilisateur peut indiquer des options de contexte sur des appels MQOPEN ou MQPUT. Pour plus d'informations sur les options de contexte, voir [Options MQOPEN concernant le contexte de message](#). Pour des descriptions des zones de descripteurs de message concernant le contexte, voir [MQMD - Descripteur de message](#).

Si vous désactivez la vérification de sécurité du contexte, n'importe quel utilisateur peut utiliser les options de sécurité autorisées par la sécurité de la file d'attente.

Vous pouvez activer ou désactiver la vérification de la sécurité du contexte au niveau du gestionnaire de files d'attente ou du groupe de partage de files d'attente.

z/OS

## Disponibilité sur z/OS

IBM MQ for z/OS propose de nombreuses fonctions pour la haute disponibilité. Cette rubrique décrit certaines des considérations en matière de disponibilité.

Plusieurs fonctions d'IBM MQ peuvent accroître la disponibilité du système si le gestionnaire de files d'attente ou l'initiateur de canal est défaillant. Pour plus d'informations sur ces fonctions, reportez-vous aux rubriques suivantes :

- [Considérations Sysplex](#)
- [Files d'attente partagées](#)
- [Canaux partagés](#)
- [Disponibilité du réseau IBM MQ](#)
- [Utilisation de z/OS Automatic Restart Manager \(ARM\)](#)
- [Utilisation de z/OS Extended Recovery Facility \(XRF\)](#)
- [Utilisation de l'attribut z/OS GROUPUR pour la récupération d'un groupe de partage de files d'attente](#)
- [Où rechercher les informations relatives à la disponibilité](#)

### Considérations Sysplex

Dans un *sysplex*, des images de système d'exploitation z/OS collaborent dans une seule image de système et communiquent à l'aide d'une unité de couplage. IBM MQ peut utiliser les fonctions de l'environnement *sysplex* pour améliorer la disponibilité.

Supprimer les affinités entre un gestionnaire de files d'attente et une image z/OS permet à un gestionnaire de files d'attente de redémarrer sur une image différente z/OS en cas d'erreur d'image.

Le mécanisme de redémarrage peut être manuel, utiliser ARM ou l'automatisation du système, si vous veillez à ce qui suit :

- Tous les ensembles de pages, les journaux, les fichiers d'amorce, les bibliothèques code et fichiers de configuration du gestionnaire de files d'attente sont définis sur des volumes partagés.
- La définition du sous-système a une portée sysplex et un nom unique au sein du sysplex.
- Le niveau du *code initial* installé sur chaque image z/OS au démarrage du système est au même niveau.
- Les adresses IP virtuelles TCP (VIPA) sont disponibles sur chaque pile TCP dans le sysplex, et vous avez configuré les programmes d'écoute et les connexions entrantes d'IBM MQ pour utiliser les adresses VIPA au lieu des noms d'hôte par défaut.

Pour plus d'informations sur l'utilisation du protocole TCP dans un sysplex, voir *TCP/IP dans un sysplex*, SG24-5235, une publication IBM Redbooks.

Par ailleurs, vous pouvez configurer plusieurs gestionnaires de files d'attente exécutés sur plusieurs images de système d'exploitation dans un sysplex, de sorte qu'ils fonctionnent comme un groupe de partage de files d'attente, qui peut profiter des files d'attente partagées et des canaux partagés pour plus de disponibilité et une charge équilibrée.

## Files d'attente partagées

Dans l'environnement du groupe de partage de files d'attente, une application peut se connecter à n'importe quel gestionnaire de files d'attente au sein du groupe de partage de files d'attente. Dans la mesure où tous les gestionnaires de files d'attente dans le groupe de partage de files d'attente peuvent accéder au même ensemble de files partagées, l'application ne dépend pas de la disponibilité d'un gestionnaire particulier ; n'importe quel gestionnaire de files d'attente dans le groupe de partage de files d'attente peut entretenir une file d'attente. Cela confère plus de disponibilité si un gestionnaire de files d'attente s'arrête car tous les autres gestionnaires dans le groupe de partage de files d'attente peuvent continuer à traiter la file d'attente. Pour plus d'informations sur la haute disponibilité des files d'attente partagées, voir [«Avantages liés à l'utilisation des files d'attente partagées»](#), à la page 197.

Pour améliorer la disponibilité des messages dans un groupe de partage de files d'attente, IBM MQ détecte si un autre gestionnaire de files d'attente du groupe se déconnecte de manière anormale de l'unité de couplage et termine les unités d'oeuvre toujours en attente pour ce gestionnaire de files d'attente lorsque c'est possible. C'est ce qu'on appelle la *reprise homologue*, décrite dans [«Reprise homologue»](#), à la page 282.

La reprise homologue ne peut récupérer les unités d'oeuvre qui étaient en attente de validation au moment de l'échec. Vous pouvez utiliser le gestionnaire ARM (Automatic Restart Manager) pour redémarrer tous les systèmes impliqués dans un échec (CICS, Db2 et IBM MQ, par exemple), et pour vous assurer qu'ils redémarrent tous sur le même nouveau processeur. Cela signifie qu'ils peuvent être resynchronisés et permet une reprise rapide des unités d'oeuvre en attente de validation. Ceci est décrit dans [«Utilisation de z/OS Automatic Restart Manager \(ARM\)»](#), à la page 298.

## Canaux partagés

Dans l'environnement du groupe de partage de files d'attente, IBM MQ fournit des fonctions qui confèrent une haute disponibilité au réseau. L'initiateur de canal permet d'utiliser des produits réseau qui équilibrent les demandes réseau sur un ensemble de serveurs éligibles et masquent les échecs de serveur au réseau (par exemple, ressources génériques VTAM). IBM MQ utilise un port générique pour les demandes entrantes afin que les demandes d'association puissent être acheminées vers un initiateur de canal disponible dans le groupe de partage des files d'attente. Ceci est décrit dans [«Canaux partagés»](#), à la page 221.

Les canaux de communications sortantes partagés prennent les messages qu'ils envoient depuis une file d'attente de transmission partagée. Les informations sur le statut d'un canal partagé se trouvent à un seul endroit pour l'intégralité du niveau du groupe de partage de files d'attente. Cela signifie qu'un canal peut être redémarré automatiquement sur un initiateur de canal différent dans le groupe de partage de files

d'attente si l'initiateur de canal, le gestionnaire de files d'attente ou le sous-système de communications échoue. C'est ce qu'on appelle la *reprise de canal homologue*, décrite dans [Canaux de sortie partagés](#).

## Disponibilité du réseau IBM MQ

Les messages IBM MQ sont transportés d'un gestionnaire de files d'attente à l'autre dans un réseau IBM MQ en utilisant des canaux. Vous pouvez changer la configuration de nombreux niveaux pour améliorer la disponibilité réseau d'un gestionnaire de files d'attente et la capacité d'un canal IBM MQ à détecter un problème de réseau et à se reconnecter.

Le protocole TCP *Keepalive* est disponible pour les canaux TCP/IP. TCP envoie donc des paquets de façon périodique entre les sessions pour détecter des défaillances du réseau. L'attribut de canal KAINTE détermine la fréquence de ces paquets pour un canal.

*AdoptMCA* permet à un canal, bloqué lors du processus de réception à la suite d'une panne du réseau, de s'arrêter et d'être remplacé par une nouvelle requête de connexion. Vous contrôlez *AdoptMCA* à l'aide de la propriété de gestionnaire de files d'attente *ADOPTMCA* et de l'utilitaire *MQSC* ou de la propriété *AdoptNewMCAType* et de l'interface *Formats de commande programmables*.

*ReceiveTimeout* empêche le blocage permanent d'un canal lors d'un appel de réception de réseau. Les paramètres de l'initiateur de canal *RCVTIME* et *RCVTMIN* déterminent les caractéristiques de délai d'attente de réception pour les canaux, comme une fonction de leur intervalle des pulsations. Pour plus d'informations, voir [Paramètres du gestionnaire de files d'attente](#).

## Utilisation de z/OS Automatic Restart Manager (ARM)

Vous pouvez utiliser IBM MQ for z/OS avec z/OS Automatic Restart Manager (ARM). Si un gestionnaire de files d'attente ou un initiateur de canal échoue, ARM le redémarre dans la même image z/OS. Si z/OS échoue, un groupe entier de sous-systèmes associés et d'applications échoue également. ARM peut redémarrer automatiquement tous les systèmes qui ont échoué, dans un ordre prédéfini, dans une autre image z/OS dans le sysplex. Il s'agit d'un redémarrage multisystème.

ARM permet une reprise rapide des transactions en attente de validation dans l'environnement de files d'attente partagées. Il confère également une haute disponibilité si vous n'utilisez pas de groupes de partage de files d'attente.

Vous devez utiliser le gestionnaire ARM pour redémarrer un gestionnaire de files d'attente dans une image z/OS différente au sein du sysplex en cas d'échec de z/OS.

Pour activer le redémarrage automatique, vous devez :

1. Configurer un fichier de couplage ARM.
2. Définir les actions de redémarrage automatique que z/OS doit exécuter dans une *règle ARM*.
3. Démarrer la règle ARM.

Si vous souhaitez redémarrer les gestionnaires de files d'attente dans différentes images z/OS, chaque gestionnaire de files d'attente dans chaque image z/OS sur laquelle le gestionnaire doit être redémarré, doit être défini avec un nom de sous-système unique de 4 caractères dans le sysplex.

L'utilisation d'ARM avec IBM MQ est décrite dans [Utilisation d'ARM dans un réseau IBM MQ](#).

## Utilisation de z/OS Extended Recovery Facility (XRF)

Vous pouvez utiliser IBM MQ dans un environnement XRF (extended recovery facility). Tous les fichiers détenus par IBM MQ (code exécutable, fichiers d'amorce, journaux et ensembles de pages) doivent être sur des périphériques de stockage à accès direct partagés entre les processeurs actifs et les processeurs XRF alternatifs.

Si vous utilisez XRF pour la récupération, vous devez arrêter le gestionnaire de files d'attente sur le processeur actif et le démarrer sur le processeur alternatif. Pour CICS, vous pouvez procéder en utilisant la table de liste des commandes fournie par CICS, ou l'opérateur du système peut le faire manuellement. Pour IMS, il s'agit d'une opération manuelle que vous devez réaliser après que la coordination du système IMS a terminé le basculement de processeur.

Les utilitaires IBM MQ doivent être terminés ou arrêtés avant que le gestionnaire de files d'attente ne soit basculé sur l'autre processeur. Ne négligez pas l'effet de cette interruption potentielle lorsque vous planifiez la récupération XRF.

Veillez à empêcher le gestionnaire de files d'attente de démarrer sur le processeur alternatif avant que le gestionnaire sur le processeur active s'arrête. Un démarrage prématuré peut entraîner de graves problèmes d'intégrité dans les données, les catalogues et le journal. La sérialisation d'accès des ressources partagées (GRS) permet d'éviter les problèmes d'intégrité en empêchant l'utilisation simultanée d'IBM MQ sur les deux systèmes. Vous devez inclure le fichier d'amorce comme ressource protégée ainsi que les processeurs actifs et XRF alternatifs dans l'anneau GRS.

## Utilisation de l'attribut z/OS GROUPUR pour la récupération d'un groupe de partage de files d'attente

Les groupes de partage de files d'attente permettent des fonctions transactionnelles supplémentaires, décrites dans cette rubrique. L'attribut GROUPUR permet que les reprises de transaction en attente de validation requises des applications client XA soient réalisées sur chaque membre du groupe de partage de files d'attente.

Si une application client XA se connecte à un groupe de partage de files d'attente via un sysplex, il est impossible de garantir à quel gestionnaire de files d'attente spécifique elle se connecte. L'utilisation de l'attribut GROUPUR par les gestionnaires de files d'attente au sein du groupe de partage de files d'attente permet l'exécution de chaque reprise de transaction en attente de validation qui est nécessaire sur chaque membre du groupe de partage de files d'attente. Même si le gestionnaire de files d'attente auquel l'application s'est connectée en premier lieu n'est pas disponible, la reprise de transaction peut avoir lieu.

Cette fonction libère l'application client XA de toute dépendance à des membres spécifiques du groupe de partage de files d'attente et élargit donc la disponibilité du gestionnaire de files d'attente. Le groupe de partage de files d'attente apparaît à l'application transactionnelle comme une entité unique fournissant toutes les fonctions d'IBM MQ et sans un seul point d'échec du gestionnaire de files d'attente.

Cette fonctionnalité n'apparaît pas à l'application transactionnelle.

## Où rechercher les informations relatives à la disponibilité

Vous pouvez trouver plus d'informations sur ces rubriques dans les sources suivantes :

<i>Tableau 23. Où rechercher les informations relatives à la disponibilité</i>	
Topic	Section
Groupes de partage de files d'attente	<a href="#">«Files d'attente partagées et groupes de partage de files d'attente», à la page 176</a>
de files d'attente partagées	<a href="#">Configuration des paramètres système</a>
Utilisation du gestionnaire de redémarrage automatique programmes utilitaires	<a href="#">Utilisation d'ARM dans un réseau IBM MQ</a>
Commandes MQSC	<a href="#">Les commandes MQSC</a>

## Surveillance et statistiques sous IBM MQ for z/OS

IBM MQ for z/OS est doté d'un ensemble de fonctions permettant de contrôler les gestionnaires de files d'attente et de collecter des statistiques.

IBM MQ est doté de fonctions permettant de contrôler le système et de collecter des statistiques. Pour plus d'informations sur ces fonctions, voir les sections suivantes :

- «Contrôle en ligne», à la page [300](#)
- «Trace IBM MQ», à la page [300](#)
- «Événements», à la page [301](#)

### Contrôle en ligne

IBM MQ contient les commandes suivantes qui permettent de contrôler le statut des objets IBM MQ :

- DISPLAY CHSTATUS affiche le statut d'un canal spécifié.
- DISPLAY QSTATUS affiche le statut d'un gestionnaire de files d'attente spécifié.
- DISPLAY CONN affiche le statut d'une connexion spécifiée.

Pour plus d'informations sur ces commandes, voir [Les commandes MQSC](#).

### Trace IBM MQ

IBM MQ est doté d'une fonction de trace permettant de collecter les informations suivantes lorsque le gestionnaire de files d'attente est en cours d'exécution :

#### Statistiques de performance

La trace de statistiques regroupe les informations suivantes afin de vous aider à contrôler la performance et ajuster votre système :

- Comptage des différentes requêtes MQI
- Comptage des différentes requêtes d'objet (statistiques du gestionnaire de files d'attente)
- Informations sur l'utilisation de Db2 (statistiques du gestionnaire Db2)
- Informations sur l'utilisation de l'unité de couplage (statistiques du gestionnaire d'unité de couplage)
- Informations sur l'utilisation du service SMDS (statistiques sur les fichiers de messages partagés)
- Informations sur l'utilisation des pools de mémoire tampon (statistiques du gestionnaire de mémoire tampon)
- Informations sur la journalisation (statistiques du gestionnaire de journaux)
- Informations sur l'utilisation du stockage (statistiques du gestionnaire de stockage)
- Informations sur les demandes de verrouillage (statistiques du gestionnaire de verrouillage)

#### Données de compte

- La trace de compte regroupe des informations sur le temps de processeur passé à traiter les appels MQI et sur le nombre de requêtes MQPUT et MQGET d'un utilisateur particulier.
- IBM MQ peut également collecter des informations sur chaque tâche en utilisant IBM MQ. Ces données sont regroupées dans un enregistrement de comptage au niveau de l'unité d'exécution. Pour chaque unité d'exécution, IBM MQ collecte également des informations sur chaque file d'attente utilisée par l'unité d'exécution.

Les données générées par la trace sont envoyées à la fonction de gestion des systèmes (fonction SMF) ou à la fonction d'analyse générale (GTF).

## Événements

Les événements IBM MQ fournissent des informations relatives aux erreurs, avertissements et autres occurrences importantes dans un gestionnaire de files d'attente. En intégrant ces événements dans votre propre application de gestion des systèmes, vous pouvez contrôler les activités sur plusieurs gestionnaires de files d'attente pour plusieurs applications IBM MQ. Vous pouvez plus particulièrement contrôler tous les gestionnaires de files d'attente dans votre système à partir d'un seul gestionnaire de files d'attente.

Les événements peuvent être signalés via un mécanisme de rapport rédigé par l'utilisateur à l'application d'administration qui prend en charge la présentation des événements à un opérateur. Les événements permettent également aux applications agissant comme agents pour d'autres réseaux d'administration, par exemple, NetView, de contrôler les rapports et de créer des alertes appropriées.

### Tâches associées

[Utilisation de la fonction de trace IBM MQ](#)

[Utilisation des événements IBM MQ](#)

z/OS

## Disposition d'une unité de récupération sur z/OS

Certaines applications transactionnelles peuvent utiliser une disposition d'unité de récupération GROUPE plutôt que QMGR lorsqu'elles sont connectées à un gestionnaire de files d'attente dans un groupe de partage de files d'attente en spécifiant le nom de groupe de partage de files d'attente plutôt que le nom du gestionnaire de files d'attente lorsqu'elles se connectent. Cela assouplit et renforce la reprise de transaction en supprimant le besoin de se reconnecter au même gestionnaire de files d'attente dans le groupe de partage de files d'attente.

Les transactions démarrées par des applications qui se sont connectées avec le nom de groupe de partage de files d'attente ont également une disposition d'unité de récupération GROUPE.

Lorsqu'une application transactionnelle se connecte avec une disposition d'unité de récupération GROUPE, elle se connecte logiquement au groupe de partage de files d'attente et n'a aucune affinité avec un gestionnaire de files d'attente particulier. Toute transaction de validation de phase 2 qui a démarré et ayant terminé la phase 1 du processus de validation, c'est-à-dire en attente de validation, peut être analysée et résolue lorsque connectée à un gestionnaire de files d'attente au sein du groupe de partage de files d'attente. Dans un scénario de reprise, cela signifie que le coordinateur de transactions n'a pas à se reconnecter au même gestionnaire de files d'attente, qui peut être indisponible à ce moment.

Les applications qui se connectent avec une disposition d'unité de récupération QMGR ont une affinité directe avec le gestionnaire de files d'attente auquel elles sont connectées. Dans un scénario de reprise, le coordinateur de transactions doit se reconnecter au même gestionnaire de files d'attente pour résoudre toutes les transactions en attente de validation, peu importe si le gestionnaire appartient à un groupe de partage de files d'attente.

Lorsque des applications indiquent un nom de groupe de partage de files d'attente et se connectent ensuite à un gestionnaire de files d'attente dans un groupe de partage de files d'attente avec une disposition d'unités de récupération GROUPE, le groupe de partage de files d'attente est logiquement un gestionnaire de ressources distinct. Cela signifie que les transactions en attente de validation ne sont visibles pour une application que si elles se reconnectent avec la même disposition d'unité de récupération. Les transactions en attente de validation d'une disposition d'unité de récupération QMGR ne sont pas visibles pour les applications qui se sont connectées avec une disposition d'unité de récupération GROUPE et vice versa.

### Concepts associés

[«Autorisation des unités de récupération GROUPE», à la page 302](#)

Un groupe de partage de files d'attente peut configurer et autoriser le support pour les unités de récupération GROUPE.

[«Prise en charge d'applications», à la page 303](#)

Cette page permet de déterminer quelles applications peuvent se connecter à une disposition des unités de récupération GROUPE

## **Autorisation des unités de récupération GROUPE**

Un groupe de partage de files d'attente peut configurer et autoriser le support pour les unités de récupération GROUPE.

Pour utiliser les unités de récupération GROUPE dans un gestionnaire de files d'attente dans un groupe de partage de files d'attente, activez l'attribut de gestionnaire de files d'attente GROUPUR. Pour plus d'informations sur ce concept, voir [«Disposition d'une unité de récupération sur z/OS»](#), à la page 301 avant de prendre connaissance du reste de cette rubrique.

Lorsque l'attribut GROUPUR est activé, le gestionnaire de files d'attente accepte les nouvelles connexions avec une disposition d'unité de récupération GROUPE. Si vous désactivez cet attribut, les nouvelles connexions avec cette disposition ne sont pas acceptées, bien que les applications déjà connectées ne sont pas affectées jusqu'à leur déconnexion.

Lorsqu'une application se connecte avec une disposition d'unité de récupération GROUPE et demande quelles transactions sont en attente de validation ou essaie de résoudre une transaction qui a été démarrée ailleurs dans le groupe de partage de files d'attente, le gestionnaire de files d'attente auquel elle est maintenant connectée doit pouvoir communiquer avec les autres membres du groupe afin de traiter la demande. Pour cela, il utilise une file d'attente partagée appelée SYSTEM.QSG.UR.RESOLUTION.QUEUE. Cette file d'attente doit se trouver dans une structure d'application récupérable appelée CSQSYSAPPL. La structure doit être récupérable car les messages persistants sont stockés dans cette file d'attente lorsqu'ils traitent des demandes de résolution.

Avant de pouvoir autoriser des unités de récupération GROUPE, vous devez vérifier que la structure de l'unité de couplage et que la file d'attente partagée sont définies. Vous pouvez utiliser les définitions de l'exemple CSQ4INSS. Lorsque la file d'attente est définie ou détectée pendant le démarrage, chaque gestionnaire de files d'attente dans le groupe de partage de files d'attente ouvre la file d'attente afin de pouvoir recevoir les demandes entrantes. Si vous souhaitez supprimer ou déplacer la file d'attente car elle a été définie de façon incorrecte, vous pouvez demander que les gestionnaires de files d'attente ferment leurs descripteurs ouverts dans la file d'attente en mettant à jour l'objet de la file d'attente afin d'interdire les demandes MQGET. Lorsque vous avez apporté les corrections nécessaires, autoriser les applications à obtenir des messages à partir de la file d'attente une nouvelle fois permet à chaque gestionnaire de files d'attente à la rouvrir. Utilisez la commande DISPLAY QSTATUS afin d'identifier quels descripteurs sont ouverts dans une file d'attente.

Une fois que vous avez terminé cette configuration, vous pouvez autoriser les unités de récupération GROUPE dans chaque gestionnaire de files d'attente auquel vous souhaitez que les applications transactionnelles puissent se connecter avec une disposition d'unité de récupération GROUPE. Nul besoin qu'il s'agisse de tous les gestionnaires de files d'attente dans le groupe de partage de files d'attente, mais si vous choisissez d'autoriser uniquement cette fonctionnalité pour un sous-ensemble du groupe de partage de files d'attente, vous devez vous assurer que vos applications essaient seulement de se connecter aux gestionnaires de files d'attente pour lesquels vous les avez autorisées. Pour plus d'informations, voir [«Prise en charge d'applications»](#), à la page 303.

Lorsque vous essayez d'activer l'attribut de gestionnaire de files d'attente GROUPUR, de nombreuses vérifications de la configuration sont effectuées. Le gestionnaire de files d'attente vérifie que :

- Il appartient à un groupe de partage de files d'attente.
- La file d'attente partagée appelée SYSTEM.QSG.UR.RESOLUTION.QUEUE a été définie, en fonction de la définition dans CSQ4INSS.
- SYSTEM.QSG.UR.RESOLUTION.QUEUE est dans une structure d'unité de couplage récupérable appelée CSQSYSAPPL.

Si l'une des vérifications ci-dessus échoue, l'attribut GROUPUR reste désactivé et un code message est renvoyé.

Ces vérifications de la configuration sont également effectuées au démarrage du gestionnaire de files d'attente si l'attribut du gestionnaire de files d'attente est activé. Si l'une des vérifications échoue au démarrage, les unités de récupération GROUPE sont désactivées et le gestionnaire de files d'attente émet un message qui identifie la vérification qui a échoué. Une fois que vous avez effectué l'action corrective nécessaire, vous devez ensuite réactiver l'attribut de gestionnaire de files d'attente.

## **Prise en charge d'applications**

Cette page permet de déterminer quelles applications peuvent se connecter à une disposition des unités de récupération GROUPE

La prise en charge de la disposition des unités de récupération GROUPE se limite à certains types d'applications transactionnelles pour lesquelles IBM MQ for z/OS est un gestionnaire de ressources, mais pas le coordinateur des transactions. Les applications transactionnelles actuellement prises en charge sont les suivantes :

- Applications client transactionnelles étendues IBM MQ
- Applications IBM MQ classes for JMS s'exécutant dans un serveur d'applications, tel que WebSphere Application Server.
- Les applications CICS s'exécutant dans CICS Transaction Server 4.2 ou une version ultérieure, si la définition de ressource CICS MQCONN est configurée avec RESYNCMEMBER (GROUPRESYNC).

### **Concepts associés**

[«Applications client transactionnelles étendues IBM MQ», à la page 303](#)

Cette page explique comment les applications client transactionnelles étendues IBM MQ peuvent utiliser l'unité de disposition de récupération GROUPE.

[«Applications CICS», à la page 304](#)

Cette page permet de déterminer comment CICS peut utiliser l'unité de disposition de récupération GROUPE.

## **Applications client transactionnelles étendues IBM MQ**

Cette page explique comment les applications client transactionnelles étendues IBM MQ peuvent utiliser l'unité de disposition de récupération GROUPE.

Une application client transactionnelle étendue, par exemple IBM MQ, est une application qui utilise JMS et s'exécute dans WebSphere Application Server en se connectant à IBM MQ sur TCP/IP et non en utilisant des liaisons locales. Ces applications client se connectent à IBM MQ for z/OS via des connexions réseau, telles que le protocole TCP/IP. Pour ces applications, il s'agit de la valeur indiquée pour le paramètre QMNAME de la chaîne xa\_info transmise dans l'appel xa\_open qui indique si une disposition d'unité de récupération QMGR ou GROUPE est utilisée. Pour plus d'informations sur xa\_open, voir [Format d'une chaîne xa\\_open](#) et [Traitement d'erreur supplémentaire pour xa\\_open](#). Pour les applications JMS, vous définissez le nom du groupe de partage de files d'attente dans la fabrique de connexions et non le nom d'un gestionnaire de files d'attente spécifique.

Afin que les applications client XA tirent profit de l'utilisation de la disposition d'unité de récupération GROUPE, vous devez configurer vos paramètres TCP/IP pour permettre à vos applications client d'être routées vers les gestionnaires de files d'attente dans le groupe de partage de files d'attente dont l'attribut GROUPUR est activé, plutôt que vers un gestionnaire de files d'attente spécifique. Vous pouvez utiliser comme technologie d'adresse IP virtuelle z/OS SysPlex Distributor, entre autres. Pour plus d'informations, voir [z/OS Communications Server et Compétences de base z/OS : adressage virtuel dynamique](#). Si vous voulez activer les unités de récupération GROUPE pour un sous-ensemble des gestionnaires de files d'attente dans votre groupe de partage de files d'attente, assurez-vous que les applications client ne peuvent pas être routées vers ceux dans lesquels elles ne sont pas activées. Il s'agit par exemple de tout gestionnaire de files d'attente dont la version est antérieure à la IBM WebSphere MQ 7.0.1.

Pour utiliser la fonction de disposition d'unités de récupération GROUPE dans une application client JMS, vous devez utiliser les bibliothèques client IBM WebSphere MQ 7.0.1 JMS. Les versions antérieures des bibliothèques client JMS utilisent la disposition d'unités de récupération du gestionnaire de files d'attente.

Vos applications client n'ont pas besoin de se connecter au groupe de partage de files d'attente à l'aide des canaux partagés.

## **Applications CICS**

Cette page permet de déterminer comment CICS peut utiliser l'unité de disposition de récupération GROUPE.

CICS 4.2 et versions ultérieures fournit l'option de resynchronisation de groupe RESYNCMEMBER (GROUPRESYNC) dans une définition de ressource MQCONN. Un CICS configuré avec cette option peut se connecter à n'importe quel gestionnaire de files d'attente adapté dans un groupe de partage de files d'attente exécuté dans la même partition logique que la région CICS. Pour prendre en charge l'option CICS GROUPRESYNC, un gestionnaire de files d'attente MQ V7.1 ou version postérieure doit s'exécuter et être activé pour la prise en charge GROUPE.

Les transactions qui s'exécutent dans une région CICS connectée à MQ à l'aide de GROUPRESYNC créent des unités d'oeuvre avec la disposition de l'unité de récupération GROUPE.

Vous pouvez utiliser RESYNCMEMBER (GROUPRESYNC) pour activer une reprise plus rapide après une défaillance du gestionnaire de files d'attente, car la région CICS peut se connecter immédiatement à un autre gestionnaire de files d'attente éligible s'exécutant sur la même partition logique, en résolvant les transactions en attente de validation si nécessaire, sans attendre le redémarrage du gestionnaire de files d'attente.

RESYNCMEMBER (GROUPRESYNC) active également des options de redémarrage plus flexibles pour CICS. Une région CICS dont la connexion MQ est configurée en vue de l'utilisation de GROUPRESYNC et des files d'attente partagées MQ peut être redémarrée dans n'importe quelle partition logique dans laquelle un gestionnaire de files d'attente s'exécute en tant que membre du même groupe de partage de files d'attente.

## **IBM MQ et autres produits z/OS**

---

Cette rubrique permet de comprendre comment IBM MQ peut collaborer avec d'autres produits z/OS.

### **Concepts associés**

[«IBM MQ et CICS», à la page 304](#)

Toutes les versions de CICS prises en charge par IBM MQ 9.0.0 et les versions ultérieures utilisent la version fournie par CICS de l'adaptateur et de la passerelle.

[«IBM MQ for z/OS et WebSphere Application Server», à la page 311](#)

Cette rubrique explique l'utilisation d'IBM MQ for z/OS par WebSphere Application Server.

### **Référence associée**

[«IBM MQ et IMS», à la page 306](#)

Cette rubrique explique le fonctionnement d'IBM MQ avec IMS. L'adaptateur IMS permet de connecter le gestionnaire de files d'attente IMS, et aux applications IMS d'utiliser l'interface MQI.

[«IBM MQ et les adaptateurs z/OS Batch, TSO et RRS», à la page 310](#)

Cette rubrique explique le fonctionnement d'IBM MQ avec les adaptateurs z/OS Batch, TSO et RRS.

## **IBM MQ et CICS**

Toutes les versions de CICS prises en charge par IBM MQ 9.0.0 et les versions ultérieures utilisent la version fournie par CICS de l'adaptateur et de la passerelle.

Pour plus d'informations sur la configuration de l'adaptateur IBM MQ CICS et des composants de IBM MQ CICS bridge, voir la section [Configuration des connexions à MQ](#) dans la documentation CICS.

### **Tâches associées**

[Utilisation d'IBM MQ avec CICS](#)

## Connexion de groupe CICS

La connexion de groupe CICS permet à une région CICS de se connecter à n'importe quel membre actif d'un groupe de partage de files d'attente IBM MQ sur la même partition logique au lieu de définir un gestionnaire de files d'attente. CICS se connecte toujours à un seul gestionnaire de files d'attente à la fois.

Deux gestionnaires de files d'attente au minimum sont nécessaires dans la partition logique pour pouvoir prendre en charge la connexion de groupe CICS. La connexion de groupe offre une meilleure disponibilité, car il n'est pas nécessaire qu'un gestionnaire de files d'attente donné soit actif. CICS se connecte à n'importe quel gestionnaire de files d'attente dans le groupe de partage de files d'attente dans la partition logique.

Pour plus d'informations, voir la documentation CICS sur la ressource MQCONN.

CICS tente de se connecter au MQNAME transmis comme s'il s'agissait d'un gestionnaire de files d'attente :

- Si le gestionnaire de files d'attente existe et qu'il est actif, la connexion fonctionne.
- Si la connexion échoue, CICS demande le statut des gestionnaires de files d'attente dans le groupe pour identifier ceux qui sont actifs dans la même partition logique.
- Si plusieurs gestionnaires de files d'attente sont actifs, CICS recherche RESYNCMEMBER (YES) et le statut de l'unité de travail pour déterminer si CICS doit se connecter à un membre particulier ou attendre s'il n'est pas actif.
- S'il n'est pas nécessaire de se connecter à un membre donné, CICS sélectionne un gestionnaire de files d'attente (en utilisant un algorithme probabiliste).
- CICS tente de se connecter au gestionnaire de files d'attente choisi.
- S'il échoue, selon le code de retour, CICS choisit le membre suivant, puis parcourt de nouveau la boucle de sélection.
- Si aucun gestionnaire de files d'attente n'est actif, CICS génère plusieurs connexions dans la liste des gestionnaires de files d'attente et attend dans ECBLIST jusqu'à ce que le premier gestionnaire de files d'attente soit disponible.

### Concepts associés

«Regroupement des unités de récupération (GROUPUR) pour CICS», à la page 305

L'attribut GROUPUR d'IBM MQ pour CICS permet une reprise homologue des unités d'oeuvre en attente de validation dans un groupe de partage de files d'attente (QSG). Un gestionnaire de files d'attente IBM MQ peut résoudre les unités en attente de validation pour le compte d'un autre gestionnaire de files d'attente dans le groupe de partage de files d'attente. Cela signifie que si CICS se reconnecte par l'intermédiaire d'une connexion de groupe à un autre gestionnaire de files d'attente du groupe de partage de files d'attente, il peut résoudre les transactions en attente de validation d'une précédente connexion IBM MQ.

### Information associée

[Prise en charge des groupes de partage de files d'attente IBM MQ](#)

## Regroupement des unités de récupération (GROUPUR) pour CICS

L'attribut GROUPUR d'IBM MQ pour CICS permet une reprise homologue des unités d'oeuvre en attente de validation dans un groupe de partage de files d'attente (QSG). Un gestionnaire de files d'attente IBM MQ peut résoudre les unités en attente de validation pour le compte d'un autre gestionnaire de files d'attente dans le groupe de partage de files d'attente. Cela signifie que si CICS se reconnecte par l'intermédiaire d'une connexion de groupe à un autre gestionnaire de files d'attente du groupe de partage de files d'attente, il peut résoudre les transactions en attente de validation d'une précédente connexion IBM MQ.

Si une région CICS travaille avec un gestionnaire de files d'attente et que le gestionnaire de files d'attente s'arrête anormalement, les transactions en attente de validation sont récupérées. Cela évite à la région CICS d'attendre le redémarrage du gestionnaire de files d'attente, et permet de résoudre les unités en attente de validation. Par conséquent, cela implique que vous devez utiliser au moins deux gestionnaires

de files d'attente dans la partition logique pour que CICS puisse se connecter à un autre gestionnaire de files d'attente en cas de fin anormale du premier gestionnaire de files d'attente.

Le nouveau paramètre RESYNCMEMBER (GROUPRESYNC) dans la définition CICS MQCONN:

- Utilise la fonction de connexion de groupe IBM MQ et la reprise homologue.
- Nécessite un gestionnaire de files d'attente avec l'attribut GROUPUR activé.
- Prend toujours en charge les paramètres CICS MQCONN RESYNCMEMBER existants (YES et NO):
  - Utilise la fonction de connexion de groupe CICS, mais pas la reprise homologue.
  - La modification des paramètres RESYNCMEMBER prend effet lors de la prochaine connexion de CICS à IBM MQ.

### Concepts associés

«Autorisation des unités de récupération GROUPE», à la page 302

Un groupe de partage de files d'attente peut configurer et autoriser le support pour les unités de récupération GROUPE.

## z/OS IBM MQ et IMS

Cette rubrique explique le fonctionnement d'IBM MQ avec IMS. L'adaptateur IMS permet de connecter le gestionnaire de files d'attente IMS, et aux applications IMS d'utiliser l'interface MQI.

La passerelle facultative IBM MQ - IMS permet aux applications d'exécuter une application IMS qui n'utilise pas l'interface MQI. Cela signifie que vous pouvez utiliser vos applications existantes avec IBM MQ sans avoir à les réécrire.

Pour plus d'informations sur ces composants, reportez-vous aux sous-rubriques suivantes :

### Concepts associés

[IMS et applications de passerelle IMS sur IBM MQ for z/OS](#)

### Tâches associées

[Configuration de l'adaptateur IMS](#)

[Configuration de la passerelle IMS](#)

[Utilisation de l'adaptateur IMS](#)

### Référence associée

[MQIIH - En-tête d'informations IMS](#)

## z/OS L'adaptateur IMS

L'adaptateur IMS est une interface entre les programmes d'application IMS et un sous-système IBM MQ.

Les adaptateurs IBM MQ permettent à plusieurs environnements d'application d'envoyer et de recevoir des messages via un réseau de mise en file d'attente des messages. L'adaptateur IMS est une interface entre les programmes d'application IMS et un sous-système IBM MQ. Il permet aux programmes d'application IMS d'utiliser l'interface MQI.

L'adaptateur TMS reçoit et interprète des requêtes pour accéder à IBM MQ à l'aide de la fonction ESAF (External Subsystem Attach Facility) fournie par IMS. Cette fonction est décrite dans le *guide de personnalisation d'IMS*. Généralement, IMS se connecte automatiquement à IBM MQ dans intervention de l'opérateur.

L'adaptateur IMS fournit un accès aux ressources IBM MQ des programmes exécutés dans les modes ou les états suivants :

- Mode Tâche (TCB - Bloc de contrôle des tâches)
- Etat du problème
- Mode mémoire-non-croisée
- Mode d'enregistrement sans accès

L'adaptateur fournit une unité d'exécution de connexion d'un bloc de contrôle des tâches d'une application à IBM MQ.

L'adaptateur prend en charge un protocole de validation en deux phases pour les changements apportés aux ressources détenues par IBM MQ avec IMS faisant office de coordinateur de point de synchronisation. Les conversations où IMS n'est pas le coordinateur de point de synchronisation, par exemple, les conversions protégées par APPC (SYNCLVL=SYNCPT), ne sont pas prises en charge par l'adaptateur IMS.

L'adaptateur fournit également une transaction de moniteur de déclenchement (CSQQTRMN). Ceci est décrit dans «Le moniteur de déclenchement IMS», à la page 307.

Vous pouvez utiliser IBM MQ avec IMS Extended Recovery Facility (XRF) pour faciliter la reprise après une erreur IMS. Pour plus d'informations sur XRF, voir le manuel *IMS Administration Guide: System*.

## Utilisation de l'adaptateur

Les programmes d'application et l'adaptateur IMS s'exécutent dans le même espace adresse. Le gestionnaire de files d'attente est séparé, dans son propre espace adresse.

Vous devez éditer les liens de chaque programme qui émet un ou plusieurs appels MQI vers un module d'interface de langue IMS approprié et, sauf s'il utilise des appels MQI dynamiques, le programme de remplacement d'API fourni par IBM MQ, CSQQSTUB. Lorsque l'application émet un appel MQI, le module de remplacement transfère le contrôle vers l'adaptateur via l'interface du sous-système externe IMS qui gère le traitement de la requête du gestionnaire de files d'attente de messages.

## Administration et utilisation du système avec IMS

Un opérateur de terminal autorisé IMS peut exécuter des commandes IMS pour contrôler et surveiller la connexion à IBM MQ. Toutefois, l'opérateur du terminal IMS ne contrôle pas l'espace adresse d'IBM MQ. Par exemple, l'opérateur ne peut pas arrêter IBM MQ depuis un espace adresse IMS.

## Restrictions

Les appels API IBM MQ suivants ne sont pas pris en charge dans une application qui utilise l'adaptateur IMS :

- MQCB
- MQCB\_FUNCTION
- MQCTL

## Le moniteur de déclenchement IMS

Le moniteur de déclenchement IMS (**CSQQTRMN**) est une application fournie par IBM MQ est une application IMS qui démarre une transaction IMS lorsqu'un événement IBM MQ se produit, par exemple, lorsqu'un message est placé dans une file d'attente donnée.

### Fonctionnement

Lorsqu'un message est inséré dans une file d'attente de message d'application, un déclenchement est généré si les conditions correspondantes sont satisfaites. Le gestionnaire de files d'attente écrit alors un message (contenant certaines données définies par l'utilisateur), le *message de déclenchement*, destiné à la file d'attente d'initiation qui a été spécifiée pour cette file d'attente de message.

Dans un environnement IMS, vous pouvez démarrer une instance de CSQQTRMN pour surveiller une file d'attente d'initialisation et y extraire les messages de déclenchement lorsqu'ils arrivent. Généralement, CSQQTRMN planifie une autre transaction IMS par une commande INSERT (ISRT) dans la file d'attente des messages IMS. L'application IMS démarrée lit le message dans la file d'attente des messages de l'application avant de la traiter. CSQQTRMN doit exécuter une persistance gérée par le bean qui n'est pas un message.

Chaque copie de CSQQTRMN sert une seule file d'attente d'initiation. Après avoir démarré, le moniteur de déclenchement s'exécute jusqu'à ce qu'IBM MQ ou IMS s'arrête.

La macro APPLCTN pour CSQQTRMN doit spécifier SCHDTYP=PARALLEL.

Dans la mesure où le moniteur de déclenchement est une persistance gérée par le bean orienté lot, les transactions IMS démarrées par le moniteur de déclenchement contiennent ce qui suit :

- Blancs dans la zone LTERM du IOPCB
- Le nom PSB de la persistance gérée par le bean du moniteur de déclenchement dans la zone ID utilisateur du IOPCB.

Si la transaction IMS cible est protégée par le serveur de sécurité (appelé auparavant RACF), il peut être nécessaire de définir CSQQTRMN comme ID utilisateur vis à vis du serveur de sécurité.

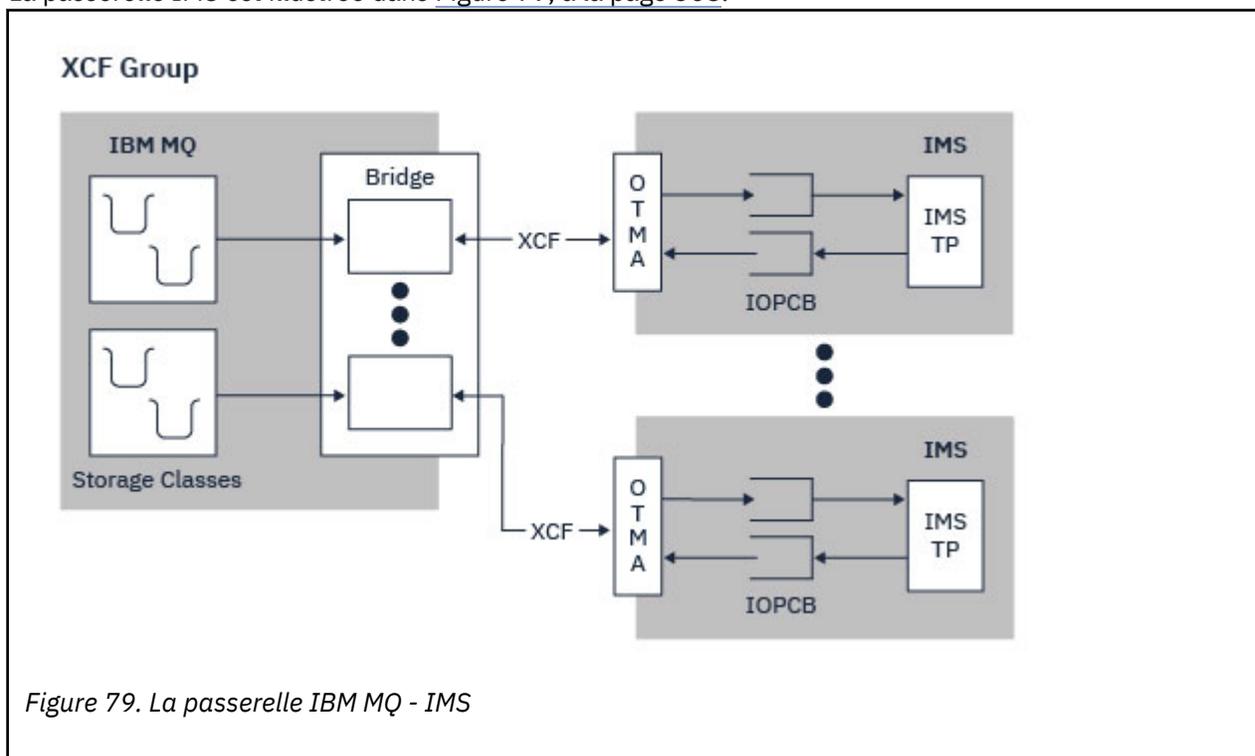
## z/OS La passerelle IBM MQ - IMS

La passerelle IBM MQ - IMS est le composant d'IBM MQ for z/OS qui permet un accès direct depuis les applications IBM MQ aux applications du système IMS.

La passerelle IBM MQ - IMS active le *support MQI implicite*. Cela implique que vous pouvez retraiter des applications existantes qui étaient contrôlées par des terminaux connectés 3270 afin qu'elles soient contrôlées à nouveau par les messages IBM MQ sans avoir à les réécrire, les recompiler ou les relier. La passerelle est un client IMS *Open Transaction Manager Access (OTMA)*.

Dans les applications de passerelle, il n'existe aucun appel IBM MQ dans l'application IMS. L'application obtient ses entrées à l'aide de GET UNIQUE (GU) vers l'IOPCB et envoie ses entrées à l'aide d'ISRT vers l'IOPCB. Les applications IBM MQ utilisent l'en-tête IMS (tel que la structure MQIIH) dans les données du message pour s'assurer que les applications peuvent s'exécuter comme elles le faisaient lorsqu'elles étaient gérées par des terminaux non programmables. Si vous utilisez une application IMS qui traite des messages multi-segments, notez que tous les segments doivent être contenus dans un message IBM MQ.

La passerelle IMS est illustrée dans Figure 79, à la page 308.



Un gestionnaire de files d'attente peut se connecter à un ou plusieurs systèmes IMS et plusieurs gestionnaires de files d'attente peuvent se connecter à un système IMS. La seule restriction est qu'ils doivent tous appartenir au même groupe XCF et se trouver dans le même sysplex.

Voir [Configuration de la passerelle IMS](#) pour des informations sur la configuration d'une passerelle IMS et l'ajout d'une connexion IMS supplémentaire au même gestionnaire de files d'attente.

## A quoi correspond OTMA ?

La fonction IMS OTMA est un protocole client-serveur sans connexion basé sur une transaction qui s'exécute sur IMS 5.1 ou version ultérieure. Il fonctionne comme une interface pour les serveurs de communications résidant sur l'hôte qui accèdent aux applications IMS TM via l'unité de couplage z/OS *Cross Systems* (XCF).

OTMA permet aux clients de se connecter à IMS pour fournir de hautes performances aux interactions entre les clients et IMS pour un grand réseau ou de nombreuses sessions. OTMA est implémenté dans un environnement sysplex z/OS. Le domaine OTMA est donc limité à celui de la fonction XCF.

## Surveillance des ressources OTMA

Le support des messages du protocole x'3C' OTMA, disponible dans IMS v10 ou version ultérieure, a été ajouté à la passerelle IBM MQ - IMS dans IBM MQ for z/OS v7.1. Ces messages sont envoyés aux clients OTMA par IMS pour signaler son état d'intégrité. Si un partenaire IMS ne peut pas traiter le volume des demandes de transactions envoyées, il indique à IBM MQ qu'un avertissement de débordement a été généré. En réponse, IBM MQ diminue la fréquence d'envoi des demandes sur la passerelle. Si IMS ne peut toujours pas traiter les demandes de transactions et qu'un débordement se produit, tous les TPIPEs vers le partenaire IMS sont suspendus. Dès que le partenaire IMS indique que le débordement ou la situation d'avertissement de débordement a disparu, IBM MQ reprend tous les TPIPEs suspendus, si nécessaire, et augmente progressivement la fréquence d'envoi des demandes de transactions jusqu'à ce que la fréquence maximale soit atteinte. Les messages de la console sont émis par IBM MQ en réponse à une modification du statut des partenaires IMS.

Si des partenaires IMS v10 sont utilisés, assurez-vous que la modification provisoire UK45082 a été appliquée.

## Envoi des transactions IMS depuis IBM MQ

Pour soumettre une transaction IMS qui utilise la passerelle, les applications insèrent des messages dans une file d'attente IBM MQ normalement. Les messages contiennent les données de transaction IMS ; elles peuvent contenir un en-tête IMS (la structure MQIIH) ou permettre à la passerelle IBM MQ - IMS d'établir des hypothèses sur les données dans le message.

IBM MQ place le message dans une file d'attente IMS (il est mis en file d'attente dans IBM MQ d'abord pour pouvoir utiliser des points de synchronisation pour garantir l'intégrité des données). La classe d'archivage de la file d'attente IBM MQ détermine si la file d'attente est une *file d'attente OTMA* (à savoir une file d'attente utilisée pour transmettre les messages à la passerelle IBM MQ - IMS), et le partenaire IMS auquel les données du message sont envoyées.

Les gestionnaires de files d'attente éloignées démarrent également des transactions IMS en écrivant dans ces files d'attente OTMA sur IBM MQ for z/OS.

Les données retournées par le système IMS sont écrites directement dans la file d'attente de réponse IBM MQ définie dans la structure de descripteur de message (MQMD). (Il peut s'agir d'une file d'attente de transmission sur le gestionnaire de files d'attente indiqué dans la zone *ReplyToQMGr* de la MQMD.)

### Concepts associés

[IMS et applications de passerelle IMS sur IBM MQ for z/OS](#)

### Tâches associées

[Personnalisation de la passerelle IMS](#)

## Référence associée

«IBM MQ et IMS», à la page 306

Cette rubrique explique le fonctionnement d'IBM MQ avec IMS. L'adaptateur IMS permet de connecter le gestionnaire de files d'attente IMS, et aux applications IMS d'utiliser l'interface MQI.

## z/OS IBM MQ et les adaptateurs z/OS Batch, TSO et RRS

Cette rubrique explique le fonctionnement d'IBM MQ avec les adaptateurs z/OS Batch, TSO et RRS.

### Présentation des adaptateurs Batch

Les adaptateurs Batch/TSO constituent l'interface entre les programmes d'application IBM MQ et z/OS exécutés sous JES, TSO ou z/OS UNIX System Services. Ces adaptateurs permettent aux programmes d'application z/OS d'utiliser l'interface MQI.

Les adaptateurs fournissent un accès aux ressources IBM MQ pour les programmes exécutés dans les modes ou les états suivants :

- Mode Tâche (TCB - Bloc de contrôle des tâches)
- Problème ou état superviseur
- Mode mémoire-non-croisée
- Mode d'enregistrement sans accès

Les connexions entre les programmes d'application et IBM MQ s'effectuent au niveau des tâches. Les adaptateurs fournissent une unité d'exécution de connexion à partir d'un bloc de contrôle de tâche d'application pour IBM MQ.

L'adaptateur Batch/TSO prend en charge un protocole de validation en une phase pour les changements apportés aux ressources détenues par IBM MQ. Il ne prend pas en charge les protocoles de validation à plusieurs phases. L'adaptateur RRS permet aux applications IBM MQ de participer à des protocoles de validation en deux phases avec des produits autorisés par RRS, coordonnés par z/OS Resource Recovery Services (RRS).

Les adaptateurs utilisent le service z/OS STIMERM pour planifier un événement asynchrone chaque seconde. Cet événement exécute un bloc de demande d'interruption (IRB) qui n'implique aucune attente par la tâche de l'application par lots. Ce bloc IRB vérifie si le bloc de contrôle d'arrêt IBM MQ a été envoyé. Si tel est le cas, le bloc IRB envoie les blocs de contrôle d'arrêt en attente d'application en attente dans un événement dans IBM MQ (par exemple, un signal ou une attente).

### L'adaptateur Batch/TSO

L'adaptateur IBM MQ Batch/TSO fournit le support IBM MQ pour les applications z/OS Batch et TSO. Tous les programmes d'application exécutés sous z/OS Batch ou TSO doivent avoir la liaison éditée CSQBSTUB du programme de remplacement d'API. Le module de remplacement fournit à l'application l'accès à tous les appels de l'interface MQI. Vous utilisez la validation et l'annulation en une phase pour les applications en émettant les appels de l'interface MQI **MQCMIT** et **MQBACK**.

### L'adaptateur RRS

*Resource Recovery Services* (RRS) est un sous-composant de z/OS qui fournit un service système pour coordonner la validation en deux phases dans les produits z/OS. L'adaptateur IBM MQ Batch/TSO RRS (adaptateur RRS) fournit le support IBM MQ pour les applications z/OS Batch et TSO qui veulent utiliser ces services. L'adaptateur RRS permet à IBM MQ de participer complètement à la coordination RRS. Les applications peuvent participer à la validation en deux phases pour d'autres produits qui prennent en charge RRS (par exemple, Db2).

L'adaptateur RRS peut fournir deux modules de remplacement ; vous devez éditer les liens les programmes d'application qui souhaitent utiliser RRS avec l'un de ces modules.

### CSQBRSTB

Ce module de remplacement vous permet d'utiliser la validation et l'annulation en deux phases pour les applications à l'aide des services de récupération des ressources d'appels RRS au lieu des appels d'interface **MQCMIT** et **MQBACK**.

Vous devez également éditer les liens le module ATRSCSS à partir de la bibliothèque SYS1.CSSLIB avec votre application. Si vous utilisez les appels d'interface MQI **MQCMIT** et **MQBACK**, vous recevrez le code de retour MQRC\_ENVIRONMENT\_ERROR.

### CSQBRRSI

Ce module de remplacement permet d'utiliser les appels MQI **MQCMIT** et **MQBACK** ; IBM MQ implémente ces appels comme appels **SRRCMIT** et **SRRBACK** RRS.

Pour plus d'informations sur la génération de programmes d'application qui utilisent l'adaptateur RRS, voir [Adaptateur batch RRS](#).

## Où trouver d'autres informations sur les adaptateurs z/OS Batch, TSO et RRS

Vous pouvez trouver plus d'informations sur les rubriques de cette section dans les sources suivantes :

Tableau 24. Où trouver d'autres informations sur l'utilisation de z/OS Batch avec IBM MQ	
Topic	Section
Configuration des adaptateurs Batch	<a href="#">Tâche 19 : Configuration des adaptateurs Batch, TSO et RRS</a>
Services de récupération des ressources pouvant être appelés par RRS	<i>Programmation MVS : Services d'appels pour les langages évolués</i>

z/OS

## IBM MQ for z/OS et WebSphere Application Server

Cette rubrique explique l'utilisation d'IBM MQ for z/OS par WebSphere Application Server.

Les applications écrites dans Java exécutées sous WebSphere Application Server peuvent utiliser Java Messaging Service (spécification JMS) pour exécuter les opérations de messagerie. La messagerie point-à-point dans cet environnement peut être fournie par un gestionnaire de files d'attente IBM MQ for z/OS.

L'utilisation d'un gestionnaire de files d'attente IBM MQ for z/OS pour fournir la messagerie offre un avantage dans la mesure où les applications JMS qui se connectent peuvent participer pleinement dans la fonctionnalité d'un réseau IBM MQ. Par exemple, elles peuvent utiliser la passerelle IMS ou échanger des messages avec les gestionnaires de files d'attente exécutés sur d'autres plateformes.

## Connexion entre WebSphere Application Server et un gestionnaire de files d'attente

Vous pouvez choisir comme classe d'objet de connexion de file d'attente *transport de client* ou *transport de liaisons*. Si vous choisissez le transport de liaisons, WebSphere Application Server et le gestionnaire de files d'attente doivent tous les deux exister dans la même image z/OS.

Notez que vous avez besoin de bibliothèques natives pour le *transport de liaisons*.

Les deux types de connexions prennent en charge les applications transactionnelles : le transport de client en utilisant les protocoles XA, le transport de liaisons en utilisant un module de remplacement WebSphere Application Server, CSQBWSTB, qui utilise les services RRS.

Pour plus d'informations sur la configuration de classes d'objet de connexion de file d'attente, voir *IBM MQ Utilisation de Java* .

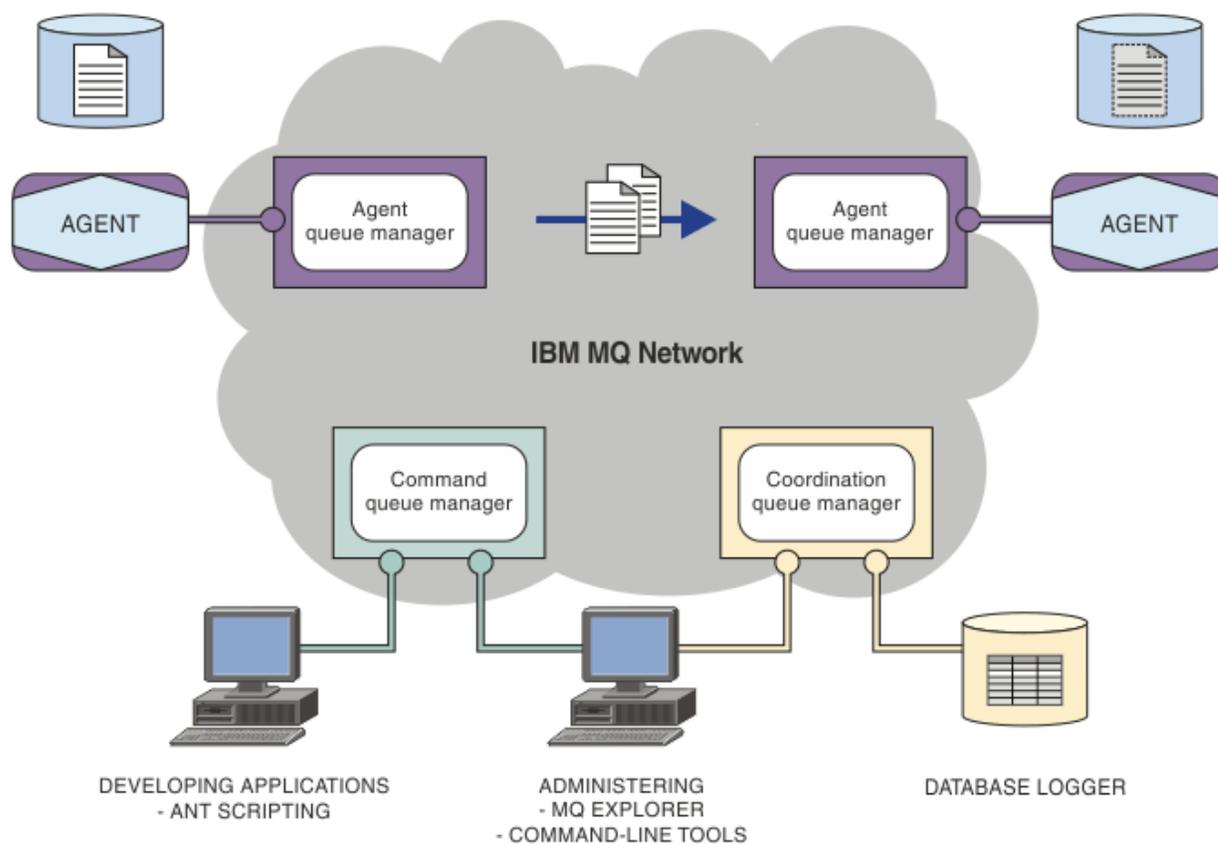
## Utilisation des fonctions IBM MQ depuis les applications JMS

Par défaut, les messages JMS dans les files d'attente IBM MQ utilisent un en-tête MQRFH2 qui contient certaines informations d'en-tête de message JMS. La plupart des applications IBM MQ existantes ne peuvent pas traiter les messages avec ces en-tête et requièrent leurs propres en-têtes, par exemple MQCIH pour CICS Bridge, ou MQWIH pour les applications IBM MQ Workflow. Pour plus de détails sur ces considérations spéciales, voir [Mapping JMS messages onto IBM MQ messages](#).

## Managed File Transfer

Managed File Transfer transfère des fichiers entre des systèmes d'une manière gérée et auditable, quelle que soit la taille des fichiers ou des systèmes d'exploitation utilisés.

Vous pouvez utiliser Managed File Transfer pour développer une solution automatisée, évolutive et personnalisée qui vous permette de gérer, d'approuver et de sécuriser les transferts de fichiers. Managed File Transfer élimine les redondances coûteuses, diminue les coûts de maintenance et maximise vos investissements informatiques existants.



Le diagramme illustre une topologie Managed File Transfer simple. Il existe deux agents, chacun se connectant à son propre gestionnaire de file d'attente d'agent sur un réseau IBM MQ. Un fichier est transféré de l'agent d'un côté du diagramme, par l'intermédiaire du réseau IBM MQ, vers l'agent de l'autre côté du diagramme. En outre, le réseau IBM MQ comporte le gestionnaire de file d'attente de coordination et un gestionnaire de file d'attente de commandes. Les applications et les outils se connectent à ces gestionnaires de files d'attente pour configurer, administrer, exploiter et consigner l'activité Managed File Transfer dans le réseau IBM MQ.

Managed File Transfer peut être installé avec quatre options différentes, en fonction de votre système d'exploitation et de la configuration générale. Ces options sont Managed File Transfer Agent, Managed File

Transfer Logger, Managed File Transfer Service et Managed File Transfer Tools. Pour plus d'informations, voir [Options du produit Managed File Transfer](#).

Vous pouvez utiliser Managed File Transfer pour effectuer les tâches suivantes :

- Créer des transferts de fichiers gérés
  -  Créer des transferts de fichiers à partir d'IBM MQ Explorer sur des plateformes Linux ou Windows.
  - Créer des transferts de fichiers à partir de la ligne de commande sur toutes les plateformes prises en charge.
  - Intégrer la fonction de transfert de fichiers dans l'outil Apache Ant.
  - Ecrire des applications qui contrôlent Managed File Transfer en plaçant les messages sur les files d'attente des commandes d'agent.
  - Planifier les transferts de fichiers à un moment ultérieur. Vous pouvez également déclencher des transferts de fichiers planifiés en fonction d'une plage d'événements du système de fichiers (par exemple, création d'un fichier).
  - Surveiller en permanence une ressource, telle qu'un répertoire, et, lorsque le contenu de cette ressource satisfait une condition prédéfinie, lancer une tâche. Il peut s'agir d'un transfert de fichiers, d'un script Ant ou d'un travail JCL.
  - Transférer des fichiers vers et depuis des files d'attente IBM MQ.
  - Transférer des fichiers vers et depuis des serveurs FTP, FTPS ou SFTP.
  - Transfert de fichiers vers et depuis des noeuds Connect:Direct.
  - Transférer des fichiers texte et binaires. Les fichiers texte sont convertis automatiquement entre les pages de code et les conventions de fin de ligne des systèmes source et cible.
  - Les transferts peuvent être sécurisés, à l'aide des normes de l'industrie pour les connexions SSL (Secure Socket Layer).
- Afficher les transferts en cours et les informations de journal sur tous les transferts sur votre réseau.
  -  Afficher le statut des transferts en cours d'IBM MQ Explorer sur des plateformes Linux ou Windows.
  -  Vérifier le statut des transferts terminés à l'aide d'IBM MQ Explorer sur des plateformes Linux ou Windows.
  - Utiliser la fonction du consignateur de base de données Managed File Transfer pour sauvegarder les messages de journal dans une base de données Db2 ou Oracle.

Managed File Transfer est développé sur IBM MQ, qui permet une distribution ponctuelle et garantie des messages entre les applications. Vous pouvez profiter des diverses fonctions d'IBM MQ. Par exemple, vous pouvez utiliser la compression des canaux pour compresser les données que vous envoyez entre les agents via les canaux IBM MQ et utiliser les canaux SSL pour sécuriser les données que vous envoyez entre les agents. Les fichiers sont transférés de manière fiable et peuvent tolérer un incident de l'infrastructure par l'intermédiaire de laquelle le transfert de fichiers est effectué. En cas de panne réseau, le transfert de fichiers redémarre à partir de là où il s'était arrêté lorsque la connectivité est restaurée.

En consolidant le transfert de fichiers avec votre réseau IBM MQ existant, vous pouvez éviter d'utiliser les ressources requises pour gérer deux infrastructures distinctes. Si vous n'êtes pas déjà un client d'IBM MQ, en créant un réseau IBM MQ pour prendre en charge Managed File Transfer, vous développez le réseau principal d'une future implémentation d'architecture SOA. Si vous êtes déjà un client d'IBM MQ, Managed File Transfer peut profiter de votre infrastructure IBM MQ et notamment d'IBM MQ Internet Pass-Thru et d'IBM Integration Bus.

Vous pouvez tirer parti des solutions à haute disponibilité IBM MQ pour améliorer la résilience de votre configuration Managed File Transfer . Si vos agents utilisent des gestionnaires de files d'attente de données répliquées (RDQM), vous devez les configurer pour qu'ils utilisent la fonction d'adresse IP

flottante. Cela signifie que les agents utilisent la même adresse IP pour communiquer avec l'une des trois instances de gestionnaire de files d'attente de données répliquées en cours d'exécution et se reconnectent automatiquement lors de la reprise en ligne (voir [Haute disponibilité des gestionnaires de files d'attente de données répliquées](#) et [Création et suppression d'une adresse IP flottante](#)). Si vous utilisez la solution de gestionnaire de files d'attente multi-instance, les applications utilisent une adresse IP différente pour communiquer avec chaque instance, qui est gérée par la reconnexion client lors de la reprise en ligne (voir [Multi-instance queue managers](#) et [Channel and client reconnection](#)).

Managed File Transfer s'intègre à un certain nombre d'autres produits IBM :

### **IBM Integration Bus**

Fichiers de processus transférés par Managed File Transfer dans le cadre d'un flux IBM Integration Bus. Pour plus d'informations, voir [Working with MFT from IBM Integration Bus](#).

### **IBM Sterling Connect:Direct**

Transférez des fichiers vers et depuis un réseau Connect:Direct existant à l'aide d'un pont Managed File Transfer Connect:Direct. Pour plus d'informations, voir [The Connect:Direct bridge](#).

### **IBM Tivoli Composite Application Manager**

IBM Tivoli Composite Application Manager fournit un agent que vous pouvez utiliser pour surveiller les informations publiées sur le gestionnaire de file d'attente de coordination.

### **Concepts associés**

Options du produit Managed File Transfer

«Présentation de la topologie MFT», à la page 315

Présentation de la manière dont les agents Managed File Transfer sont connectés au gestionnaire de file d'attente de coordination dans un réseau IBM MQ.

«Fonctionnement de MFT avec IBM MQ», à la page 314

Managed File Transfer interagit de diverses façons avec IBM MQ.

## **Fonctionnement de MFT avec IBM MQ**

Managed File Transfer interagit de diverses façons avec IBM MQ.

- Managed File Transfer transfère des fichiers entre des processus d'agent en divisant chaque fichier en un ou plusieurs messages et en transmettant les messages sur votre réseau IBM MQ.
- Les processus d'agent déplacent les données de fichier à l'aide de messages non persistants afin de limiter l'impact sur vos journaux IBM MQ. En communiquant les uns avec les autres, les processus d'agent régulent le flux de messages contenant des données de fichier. Ainsi, les messages contenant des données de fichier ne s'accumulent pas dans les files d'attente de transmission IBM MQ et au cas où un message non persistant ne serait pas distribué, les données de fichier sont envoyées à nouveau.
- Les agents Managed File Transfer utilisent plusieurs files d'attente IBM MQ. Pour plus d'informations, voir [MFT system queues and the system topic](#).
- Bien que certaines de ces files d'attente soient strictement réservées à un usage interne, un agent peut accepter des demandes sous forme de messages de commande formatés spécialement, qui sont envoyés dans une file d'attente spécifique lue par l'agent. Les commandes de ligne de commande et le plug-in IBM MQ Explorer envoient des messages IBM MQ à l'agent pour l'inviter à effectuer l'action voulue. Vous pouvez écrire des applications IBM MQ qui interagissent avec l'agent de cette façon. Pour plus d'informations, voir [Controlling MFT by putting messages on the agent command queue](#).
- Les agents Managed File Transfer envoient des informations sur leur état et la progression et le résultat des transferts vers un gestionnaire de files d'attente MQ qui a été désigné comme gestionnaire de file d'attente de coordination. Ces informations sont publiées par le gestionnaire de file d'attente de coordination et les applications souhaitant surveiller la progression des transferts ou conserver des enregistrements des transferts effectués peuvent s'y abonner. Les commandes de ligne de commande et le plug-in IBM MQ Explorer peuvent utiliser les informations qui sont publiées. Vous pouvez écrire des applications IBM MQ qui utilisent ces informations. Pour plus d'informations sur la rubrique dans laquelle les informations sont publiées, voir [SYSTEM.FTE](#).

- Les composants principaux de Managed File Transfer bénéficient de la capacité des gestionnaires de files d'attente IBM MQ à stocker et retransmettre des messages. En d'autres termes, en cas d'indisponibilité, les parties non affectées de votre infrastructure peuvent continuer de transférer des fichiers. Cette possibilité s'étend au gestionnaire de file d'attente de coordination, où une combinaison de stockage et de retransmission et d'abonnements durables permet au gestionnaire de file d'attente de coordination d'être indisponible sans que des informations essentielles sur les transferts de fichiers qui ont eu lieu ne soient perdues.

## Présentation de la topologie MFT

Présentation de la manière dont les agents Managed File Transfer sont connectés au gestionnaire de file d'attente de coordination dans un réseau IBM MQ.

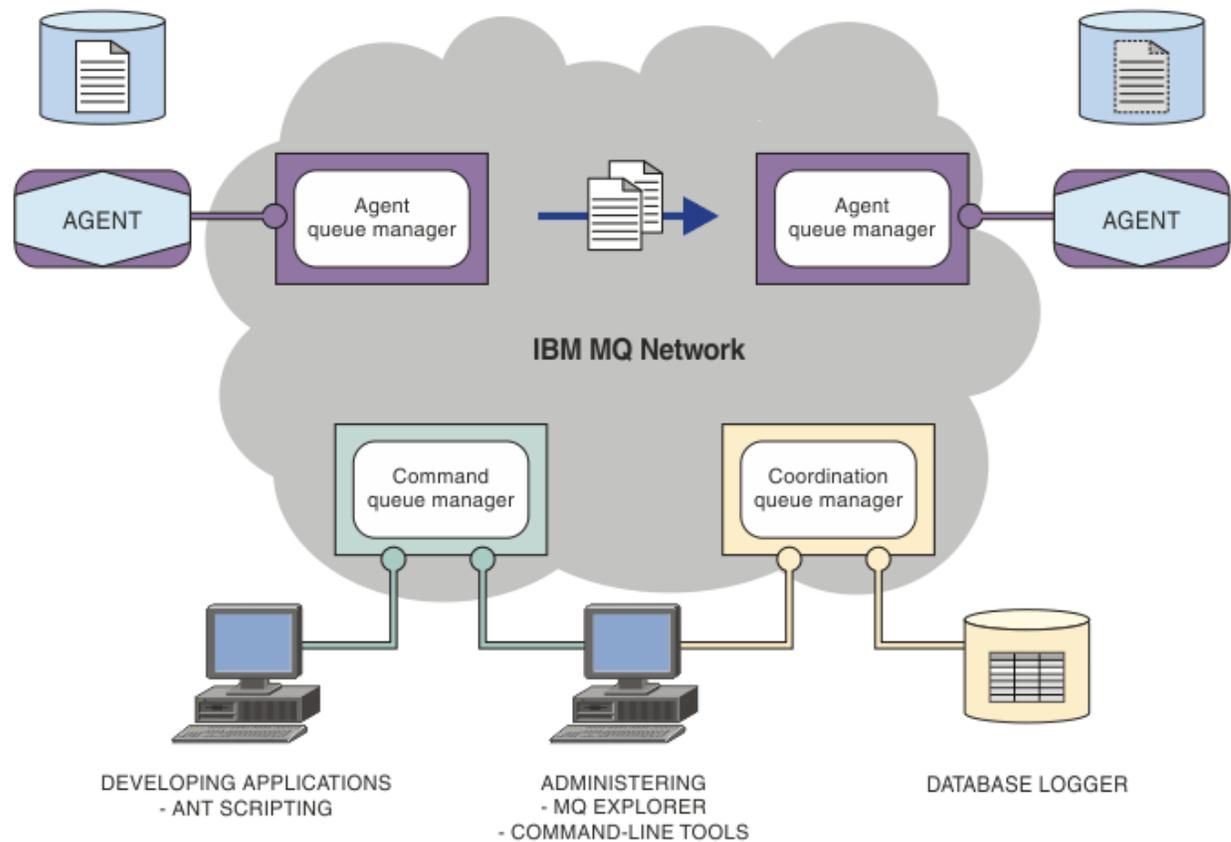
Les agents Managed File Transfer envoient et reçoivent les fichiers transférés. Chaque agent possède son propre ensemble de files d'attente sur son gestionnaire de files d'attente associé et l'agent est connecté à son gestionnaire de files d'attente en mode liaisons ou client. Il peut également utiliser le gestionnaire de file d'attente de coordination comme gestionnaire de files d'attente.

Le gestionnaire de file d'attente de coordination publie les informations d'audit et de transfert de fichier. Il représente un point unique pour la collecte des informations sur l'agent, le statut de transfert et l'audit de transfert. Il n'est pas nécessaire que le gestionnaire de file d'attente de coordination soit disponible pour que les transferts aient lieu. Si le gestionnaire de file d'attente de coordination devient temporairement indisponible, les transferts continuent normalement. Les messages de statut et d'audit sont stockés dans les gestionnaires de files d'attente des agents jusqu'à ce que le gestionnaire de file d'attente de coordination devienne disponible et puisse les traiter normalement.

Les agents s'enregistrent auprès du gestionnaire de file d'attente de coordination et publient les informations détaillées les concernant dans ce dernier. Les informations sur les agents sont utilisées par le plug-in Managed File Transfer pour permettre le démarrage des transferts depuis IBM MQ Explorer. Les informations sur les agents collectées sur le gestionnaire de file d'attente de coordination sont également utilisées par les commandes pour afficher les informations sur les agents et le statut des agents.

Les informations sur le statut de transfert et l'audit de transfert sont publiées sur le gestionnaire de file d'attente de coordination. Les informations sur le statut de transfert et l'audit de transfert sont utilisées par le plug-in Managed File Transfer pour surveiller la progression des transferts à partir d'IBM MQ Explorer. Les informations sur l'audit de transfert stockées dans le gestionnaire de file d'attente de coordination peuvent être conservées à des fins d'audit.

Le gestionnaire de file d'attente de commandes permet de se connecter au réseau IBM MQ ; il s'agit du gestionnaire de files d'attente auquel vous vous connectez lorsque vous exécutez des commandes Managed File Transfer.



### Concepts associés

«Managed File Transfer», à la page 312

Managed File Transfer transfère des fichiers entre des systèmes d'une manière gérée et auditable, quelle que soit la taille des fichiers ou des systèmes d'exploitation utilisés.

«Fonctionnement de MFT avec IBM MQ», à la page 314

Managed File Transfer interagit de diverses façons avec IBM MQ.

### Référence associée

[Scénario pour Managed File Transfer](#)

## Présentation de MFT REST API

L'REST API prend en charge certaines commandes Managed File Transfer, notamment le listage des transferts, et des détails sur les agents de transfert de fichier.

Depuis IBM MQ 9.1.0, l'REST API inclut des options permettant de répertorier tous les transferts Managed File Transfer en cours et d'interroger le statut des agents Managed File Transfer. Pour plus d'informations, voir [Initiation à l'REST API MFT](#).

## IBM MQ Internet Pass-Thru

IBM MQ Internet Pass-Thru (MQIPT) est un composant facultatif d'IBM MQ que vous pouvez utiliser pour implémenter des solutions de messagerie entre des sites distants sur Internet.

**V 9.2.0** Depuis IBM MQ 9.2.0, MQIPT est un composant facultatif d'IBM MQ. Pour obtenir les fichiers d'installation MQIPT pour IBM MQ 9.2.x, accédez à [IBM Fix Central pour IBM MQ](#). Avant IBM MQ 9.2.0, MQIPT était disponible sous forme de package de prise en charge.

Il n'est pas nécessaire d'exécuter IBM MQ 9.2.0 pour pouvoir utiliser MQIPT dans IBM MQ 9.2.0 ou version ultérieure. Vous pouvez utiliser MQIPT pour connecter n'importe quelle version prise en charge

de IBM MQ, et vous n'avez pas besoin d'installer d'autres composants IBM MQ à la même version que MQIPT.

Si vous avez acheté l'autorisation d'utilisation d'IBM MQ, vous pouvez installer autant de copies de MQIPT que nécessaire. Les installations de MQIPT ne sont pas comptabilisées dans le cadre de votre autorisation d'utilisation d'IBM MQ. Pour plus d'informations sur l'octroi de licence IBM MQ, voir [Informations sur les licences IBM MQ](#).

**Remarque :** Cette documentation concerne MQIPT dans IBM MQ 9.2.0 et les versions ultérieures. Pour la documentation du package de prise en charge de MQIPT (version 2.1) dans l'IBM Documentation, voir [MQIPT \(SupportPac MS81\)](#) dans la documentation IBM MQ 9.0.

**Remarque :** Si vous utilisez MQIPT version 2.1 ou antérieure, vous êtes invité à effectuer une mise à niveau vers MQIPT pour IBM MQ 9.2.x, car la date de fin de support pour le module de prise en charge MQIPT est le 30 septembre 2020.

IBM MQ Internet Pass-Thru s'exécute en tant que service autonome pouvant recevoir et transmettre des flux de messages IBM MQ, entre deux gestionnaires de files d'attente IBM MQ ou entre un client IBM MQ et un gestionnaire de files d'attente IBM MQ.

MQIPT active cette connexion lorsque le client et le serveur ne se trouvent pas sur le même réseau physique.

Une ou plusieurs instances de MQIPT peuvent être placées dans le chemin d'accès de communication entre deux gestionnaires de files d'attente IBM MQ, ou entre un client IBM MQ et un gestionnaire de files d'attente IBM MQ. Les instances de MQIPT permettent aux deux systèmes IBM MQ d'échanger des messages sans avoir besoin d'une connexion TCP/IP directe entre les deux systèmes. Ceci est utile si la configuration de pare-feu interdit une connexion TCP/IP directe entre les deux systèmes.

MQIPT écoute les connexions entrantes sur un ou plusieurs ports TCP/IP : messages IBM MQ normaux, messages IBM MQ tunnelisés dans HTTP, ou messages chiffrés avec TLS (Transport Layer Security) ou SSL (Secure Sockets Layer). MQIPT peut gérer plusieurs connexions simultanées.

Le canal IBM MQ qui établit la demande de connexion TCP/IP initiale est désigné comme l'*appelant*. Le canal auquel il tente de se connecter est le *répondeur*, et le gestionnaire de files d'attente qu'il tente de contacter est le *gestionnaire de files d'attente de destination*.

MQIPT conserve des données en mémoire lorsqu'il les transmet de la source vers la destination. Les données ne sont pas enregistrées sur le disque (à l'exception de la mémoire paginée sur le disque par le système d'exploitation). La seule fois où MQIPT accède explicitement au disque est lorsqu'il lit le fichier de configuration et écrit les enregistrements de trace et du journal de connexion.

La gamme complète des types de canaux IBM MQ est disponible via une ou plusieurs instances de MQIPT. La présence de MQIPT dans un chemin de communication n'a aucun effet sur les caractéristiques fonctionnelles des composants IBM MQ connectés, mais peut impacter les performances du transfert de messages.

MQIPT peut être utilisé en conjonction avec IBM MQ et IBM Integration Bus, comme décrit dans [«Configurations possibles de MQIPT»](#), à la page 321.

Pour installer MQIPT, voir [Installation de MQIPT](#).

#### **Tâches associées**

[Configuration de IBM MQ Internet Pass-Thru](#)

[Administration et configuration d'IBM MQ Internet Pass-Thru](#)

#### **Référence associée**

[IBM MQ Internet Pass-Thru Référence de configuration](#)

## **Utilisations de MQIPT**

Potentiellement, vous pouvez utiliser IBM MQ Internet Pass-Thru (MQIPT) de nombreuses façons.

## MQIPT peut être utilisé comme concentrateur de canal

Si vous utilisez MQIPT de cette façon, les canaux en provenance ou à destination de plusieurs hôtes distincts sont perçus par un pare-feu comme s'ils étaient tous originaires ou à destination du même hôte MQIPT. Il est ainsi plus facile de définir et de gérer les règles de filtrage des pare-feux.

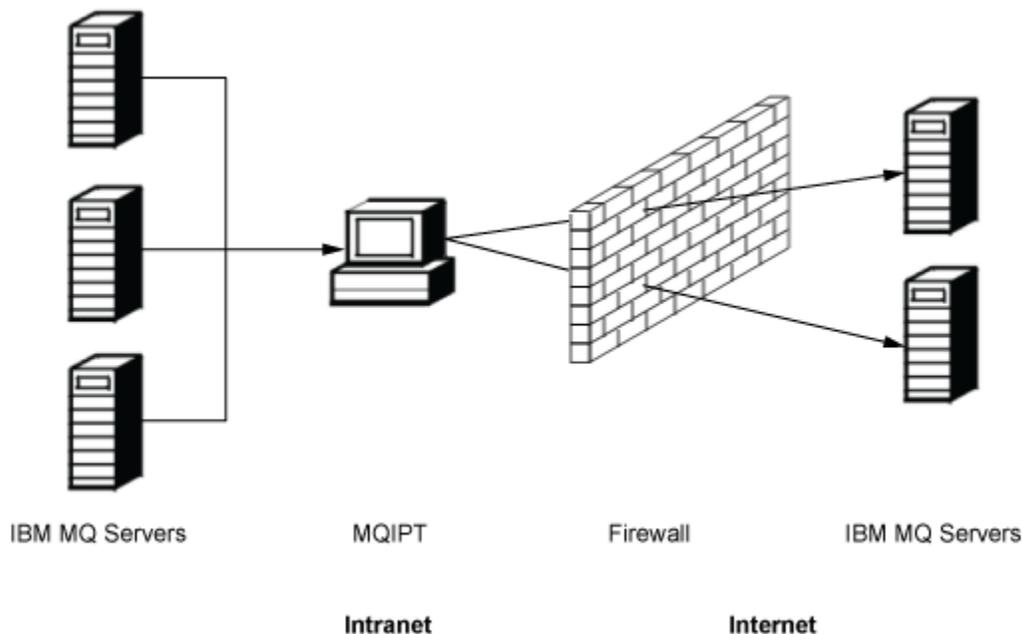


Figure 80. Exemple d'instance MQIPT utilisée comme concentrateur de canal

## MQIPT peut être placé dans une zone démilitarisée pour fournir un seul point d'accès

Si MQIPT est placé à l'intérieur d'un pare-feu dans une zone démilitarisée (il s'agit d'une configuration de pare-feu permettant de sécuriser les réseaux locaux), sur un ordinateur utilisant une adresse IP connue et sécurisée, MQIPT peut être utilisé pour écouter les connexions de canal entrantes IBM MQ. Il peut alors les transmettre à l'intranet sécurisé. Le pare-feu interne doit autoriser cet ordinateur sécurisé à établir des connexions entrantes. Dans cette configuration, MQIPT empêche les demandes externes de recevoir les vraies adresses IP des ordinateurs qui se trouvent sur l'intranet sécurisé. MQIPT fournit ainsi un seul point d'accès.

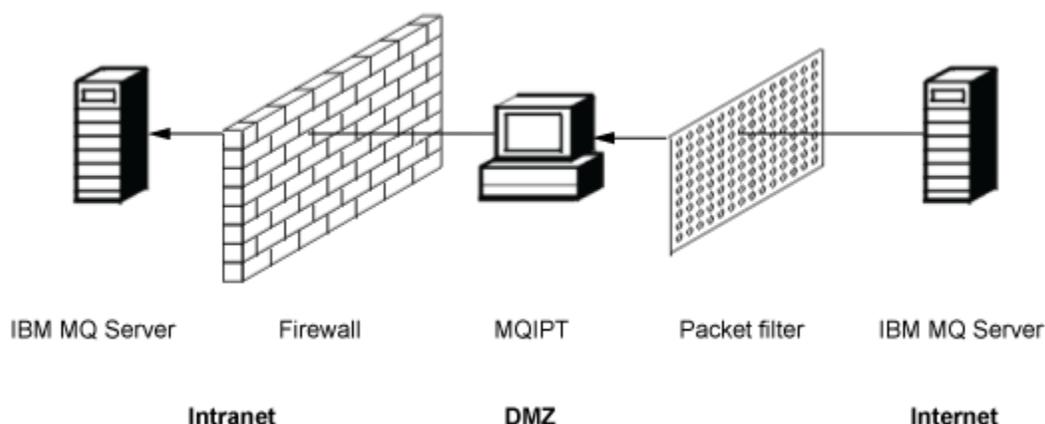


Figure 81. Exemple d'instance MQIPT dans un pare-feu de zone démilitarisée

### MQIPT peut communiquer via la tunnellation HTTP

Si deux instances de MQIPT sont déployées en ligne, elles peuvent communiquer via HTTP. La fonction de tunnellation HTTP permet aux demandes d'être transmises via des pare-feux, par l'intermédiaire des proxys HTTP existants. Le premier MQIPT insère le protocole IBM MQ dans HTTP et le second extrait le protocole IBM MQ à partir de son encapsuleur HTTP et le transmet au gestionnaire de files d'attente de destination.

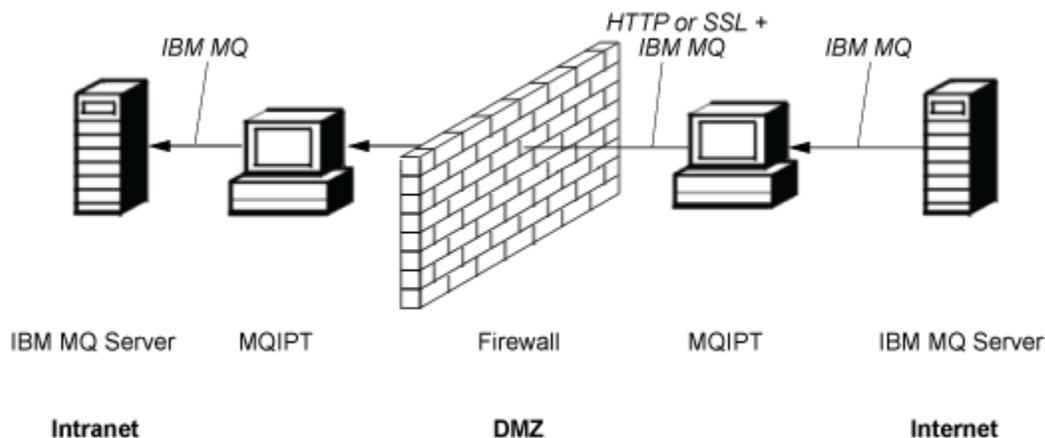


Figure 82. Exemple d'instance de MQIPT avec tunnellation HTTP

### MQIPT peut chiffrer les messages

Si MQIPT est configuré conformément à l'exemple précédent, les demandes peuvent être chiffrées avant leur transmission via des pare-feux. La première instance de MQIPT chiffre les données et la deuxième les déchiffre à l'aide du protocole SSL/TLS avant de les envoyer au gestionnaire de files d'attente de destination.

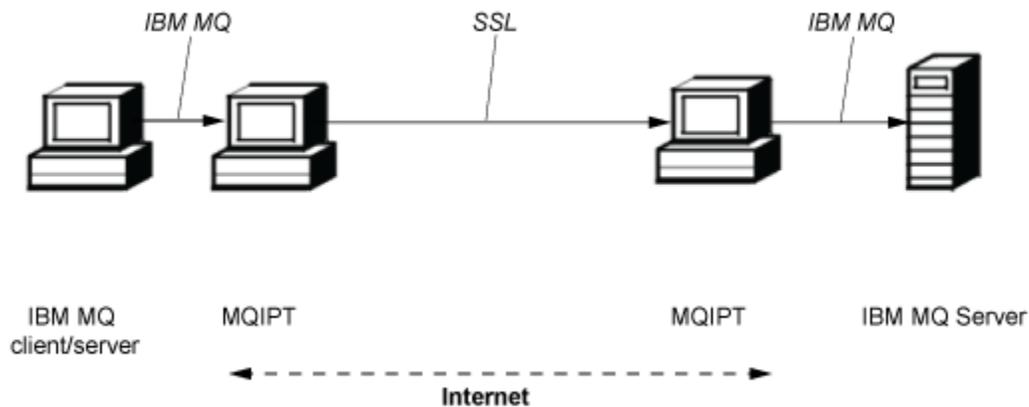


Figure 83. Exemple de MQIPT avec SSL/TLS

## Fonctionnement de MQIPT

Dans sa configuration la plus simple, MQIPT fait office de réexpéditeur de protocole IBM MQ. Il est à l'écoute sur un port TCP/IP et accepte les demandes de connexions provenant des canaux IBM MQ.

Si une demande syntaxiquement correcte est reçue, MQIPT établit une autre connexion TCP/IP entre lui-même et le gestionnaire de files d'attente IBM MQ. Il transmet alors tous les paquets de protocole qu'il reçoit de sa connexion entrante au gestionnaire de files d'attente de destination, et renvoie les paquets de protocole provenant du gestionnaire de files d'attente de destination à la connexion entrante d'origine.

Aucune modification du protocole IBM MQ (client/serveur ou gestionnaire de files d'attente vers gestionnaire de files d'attente) n'est nécessaire, car aucune des deux extrémités n'est directement informée de la présence de l'intermédiaire. De nouvelles versions du code client ou serveur d'IBM MQ ne sont pas requises.

Pour utiliser MQIPT, le canal appelant doit être configuré pour utiliser le nom d'hôte et le port de MQIPT, et non le nom d'hôte et le port du gestionnaire de files d'attente de destination. La propriété **CONNNAME** du canal IBM MQ est utilisée. MQIPT lit les données entrantes et se contente de les transmettre au gestionnaire de files d'attente de destination. Les autres zones de configuration, telles que l'ID utilisateur et le mot de passe dans un canal client/serveur, sont également transmises au gestionnaire de files d'attente de destination.

## Utilisation de plusieurs gestionnaires de files d'attente

MQIPT peut être utilisé pour autoriser l'accès à plusieurs gestionnaires de files d'attente de destination. Dans ce cas, un mécanisme indiquant à MQIPT à quel gestionnaire de files d'attente il doit se connecter doit exister, pour que MQIPT puisse utiliser le numéro de port TCP/IP entrant pour déterminer à quel gestionnaire de files d'attente se connecter.

Par conséquent, vous pouvez configurer MQIPT pour qu'il soit à l'écoute sur plusieurs ports TCP/IP. Chaque port d'écoute est mappé à un gestionnaire de files d'attente de destination via une MQIPT route. Vous pouvez définir jusqu'à 100 routes, qui associent un port d'écoute TCP/IP au nom d'hôte et au port du gestionnaire de files d'attente de destination. Cela signifie que le nom d'hôte (adresse IP) du gestionnaire de files d'attente de destination n'est jamais visible pour le canal émetteur. Chaque route peut prendre en charge plusieurs connexions entre son port d'écoute et sa destination, chaque connexion agissant de façon indépendante.

## Fichier de configuration MQIPT

MQIPT utilise un fichier de configuration appelé `mqipt.conf`. Ce fichier contient les définitions de toutes les routes et leurs propriétés associées. Pour plus d'informations sur `mqipt.conf`, voir [Administration et configuration de IBM MQ Internet Pass-Thru](#).

Lorsque MQIPT est lancé, il démarre chaque route répertoriée dans le fichier de configuration. Les messages sont écrits dans la console système indiquant le statut de chaque route. Lorsque MQCPI078 s'affiche pour une route, cela signifie que cette route est prête à accepter des demandes de connexion.

## Configurations possibles de MQIPT

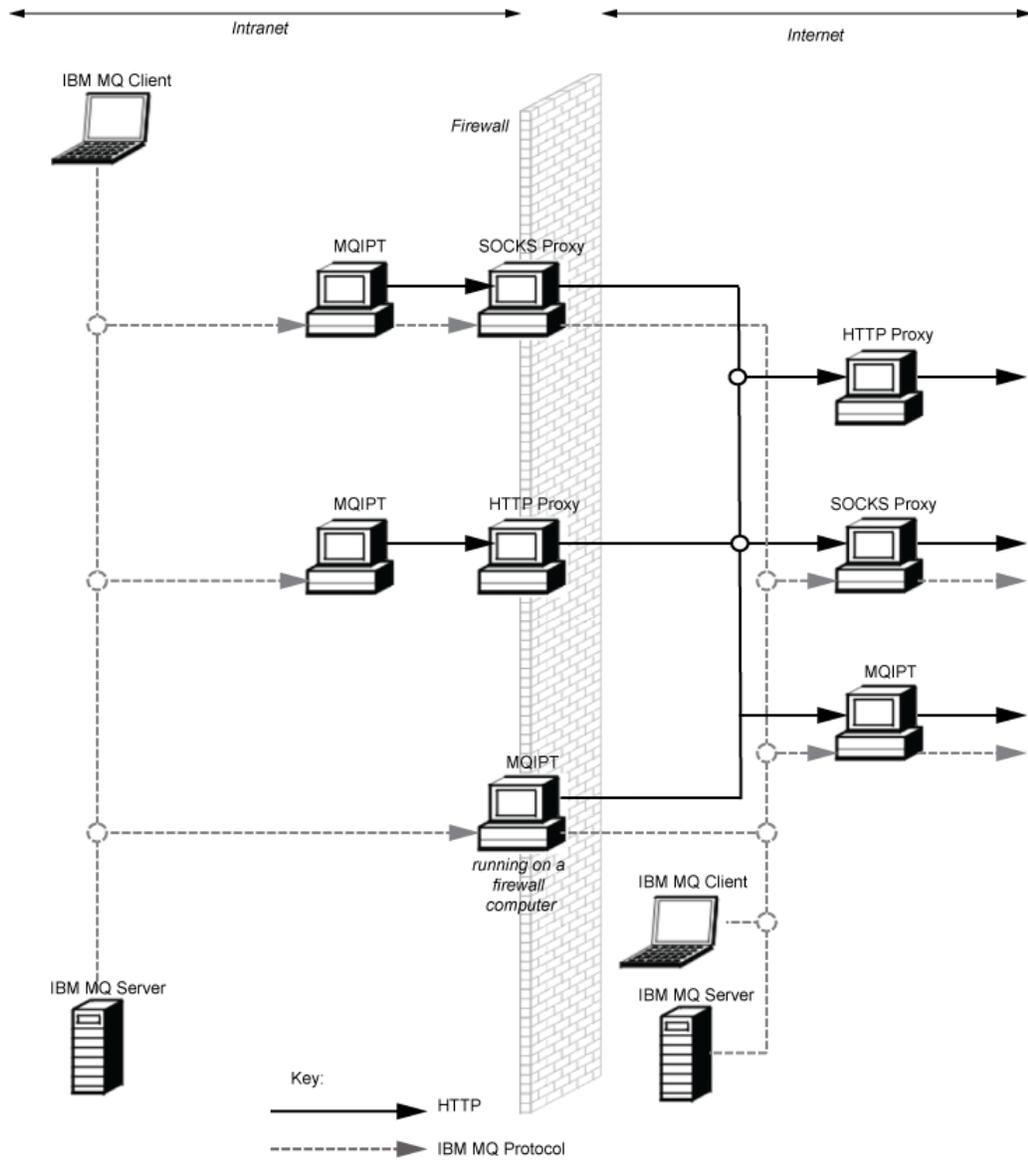
MQIPT peut être utilisé avec IBM MQ et IBM Integration Bus.

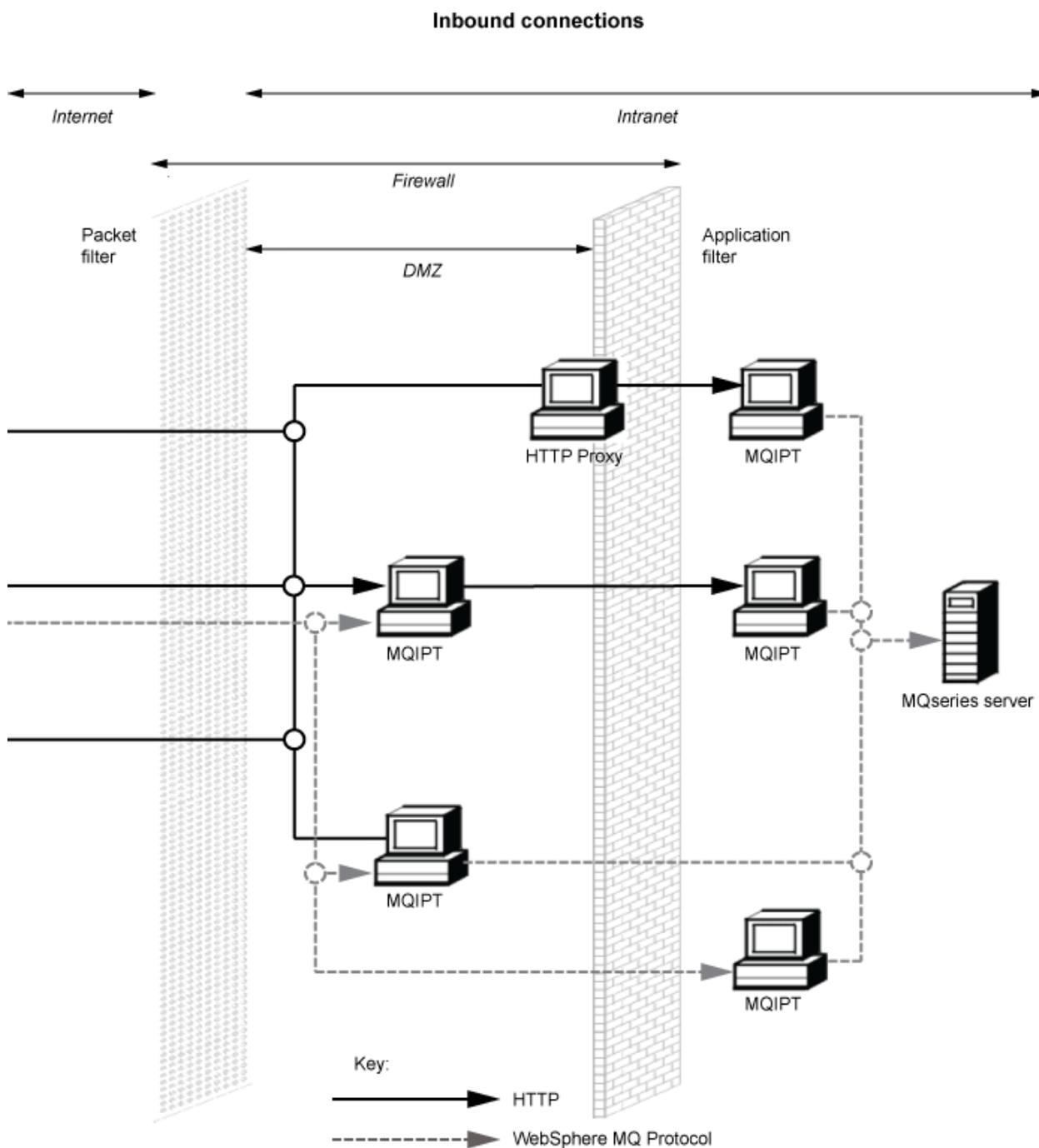
La figure ci-après en plusieurs parties représente de nombreuses configurations possibles de MQIPT dans une topologie IBM MQ. Elle illustre les différentes méthodes d'envoi de messages utilisées par MQIPT. Elle représente les clients et les serveurs sur un réseau intranet, à l'intérieur d'un pare-feu et sur Internet en dehors du pare-feu, transmettant des messages à MQIPT, au proxy HTTP ou au proxy SOCKS, qui les retransmettent.

Les messages sont reçus par un proxy MQIPT ou un proxy HTTP dans une zone démilitarisée avant leur transmission à un serveur via le pare-feu entrant.

Notez que le proxy HTTP, le proxy SOCKS, et les ordinateurs MQIPT côté intranet du pare-feu représentent la possibilité de chaîner plusieurs ordinateurs sur Internet. Par exemple, un ordinateur MQIPT pourra communiquer via un ou plusieurs ordinateurs proxy SOCKS ou HTTP, ou d'autres ordinateurs MQIPT, avant d'atteindre sa cible.

### Outbound connections





## Configurations compatibles

Scénarios de connexion compatibles où un client ou un gestionnaire de files d'attente IBM MQ communique avec MQIPT. La même route ou une seconde route MQIPT est utilisée pour communiquer avec un gestionnaire de files d'attente de destination.

## Configurations compatibles avec une seule route MQIPT

Vous pouvez utiliser une seule route MQIPT pour communiquer avec IBM MQ.

Les colonnes du [Tableau 25](#), à la page 324 contiennent les informations suivantes :

1. Le protocole utilisé entre IBM MQ et la route MQIPT. La connexion peut être créée par un client ou un gestionnaire de files d'attente IBM MQ, et peut utiliser les formats et protocoles IBM MQ (FAP) ou un protocole SSL/TLS.
2. Mode dans lequel fonctionne la route MQIPT. Le format de la communication sur Internet entre MQIPT et IBM MQ est déterminé par la configuration de la route MQIPT. Notez que lorsque la table mentionne SSL, vous pouvez également utiliser TLS.
3. Le protocole utilisé entre la route MQIPT et le gestionnaire de files d'attente de destination.

<b>1. IBM MQ Protocole source</b>	<b>2. Mode de la route MQIPT</b>	<b>3. IBM MQ Protocole cible</b>
FAP	Proxy FAP (par défaut)	FAP
	Serveur FAP et client SSL	SSL/TLS
SSL/TLS	Proxy SSL	SSL/TLS
	Serveur SSL et client FAP	FAP
	Serveur SSL et client SSL	SSL/TLS

## Configurations compatibles avec plusieurs routes MQIPT

Vous pouvez choisir d'utiliser plusieurs routes, sur une ou plusieurs instances de MQIPT, afin de communiquer avec IBM MQ.

Les colonnes du [Tableau 26](#), à la page 325 contiennent les informations suivantes :

1. Le protocole utilisé entre IBM MQ et la première route MQIPT. La connexion peut être créée par un client ou un gestionnaire de files d'attente IBM MQ, et peut utiliser les formats et protocoles IBM MQ (FAP) ou un protocole SSL/TLS.
2. Mode dans lequel fonctionne la première route MQIPT. Le format de la communication sur Internet entre MQIPT et IBM MQ est déterminé par la configuration de la route MQIPT. Notez que lorsque la table mentionne SSL, vous pouvez également utiliser TLS.
3. Mode dans lequel fonctionne la seconde route MQIPT.
4. Le protocole utilisé entre la seconde route MQIPT et le gestionnaire de files d'attente de destination.

Tableau 26. Configurations valides avec plusieurs instances de MQIPT

1. IBM MQ Protocole source	2. Mode de la première route MQIPT	3. Mode de la seconde route MQIPT	4. IBM MQ Protocole cible
FAP (par défaut)	Proxy FAP (par défaut)	Proxy FAP (par défaut)	FAP
	Serveur FAP et client SSL	Proxy SSL	SSL/TLS
		Serveur SSL et client FAP	FAP
		Serveur SSL et client SSL	SSL/TLS
	Client HTTP	Serveur HTTP et client SSL	SSL/TLS
	Client HTTPS	Serveur HTTPS et client SSL	SSL/TLS
	Client HTTP	Serveur HTTP	FAP
	Client HTTPS	Serveur HTTPS	FAP
SSL/TLS	Proxy SSL	Proxy SSL	SSL/TLS
		Serveur SSL et client FAP	FAP
		Serveur SSL et client SSL	SSL/TLS
	Client HTTP	Serveur HTTP	FAP
	Client HTTPS	Serveur HTTPS	SSL/TLS
	Client HTTP	Serveur HTTP et client SSL	FAP
	Client HTTPS	Serveur HTTPS et client SSL	SSL/TLS

## Configurations de canal prises en charge

Tous les types de canaux IBM MQ sont pris en charge, mais la configuration est limitée aux connexions TCP/IP. Pour un client ou un gestionnaire de files d'attente IBM MQ, MQIPT est présenté comme gestionnaire de files d'attente de destination. Lorsque la configuration de canal requiert un hôte de destination et un numéro de port, le nom d'hôte et le numéro de port d'écoute de MQIPT sont indiqués.

### Canaux client/serveur

MQIPT écoute les demandes de connexion client entrantes, puis les transmet via la tunnellation HTTP, via SSL/TLS, ou en tant que paquets de protocole IBM MQ standard. Si MQIPT utilise la tunnellation HTTP ou SSL/TLS, il transmet les demandes via une connexion à une autre instance de MQIPT. S'il n'utilise pas la tunnellation HTTP, il transmet les demandes via une connexion à ce qu'il considère comme le gestionnaire de files d'attente de destination (bien que celui-ci puisse être à son tour une autre instance de MQIPT). Une fois que le gestionnaire de files d'attente de destination a accepté la connexion client, les paquets sont relayés entre le client et le serveur.

### Canaux émetteur/récepteur de cluster

Si MQIPT reçoit une demande entrante d'un canal émetteur de cluster, il considère que le gestionnaire de files d'attente prend en charge SOCKS et que la vraie adresse de destination sera obtenue au cours de l'établissement de liaison de SOCKS. Il transmet la demande à l'instance de MQIPT suivante ou au gestionnaire de files d'attente de destination exactement de la même manière que pour les canaux de connexion client. Il en va de même pour les canaux émetteurs de cluster définis automatiquement.

### **Emetteur/récepteur**

Si MQIPT reçoit une demande entrante d'un canal émetteur, il la transmet à l'instance de MQIPT suivante ou au gestionnaire de files d'attente de destination exactement de la même manière que pour les canaux de connexion client. Le gestionnaire de files d'attente de destination valide la demande entrante et démarre le canal récepteur, si besoin est. Toutes les communications entre les canaux émetteur et récepteur (y compris les flux de sécurité) sont relayées.

### **Demandeur/serveur**

Cette combinaison est traitée de la même manière que les configurations précédentes. La validation de la demande de connexion est effectuée par le canal serveur sur le gestionnaire de files d'attente de destination.

### **Demandeur/émetteur**

La configuration de "rappel" est utile si les deux gestionnaires de files d'attente ne sont pas autorisés à se connecter directement l'un à l'autre, mais que tous les deux sont autorisés à se connecter à MQIPT et à accepter les connexions provenant de celui-ci.

### **Serveur/demandeur et serveur/récepteur**

Ils sont gérés par MQIPT de la même manière qu'il gère la configuration Sender/Receiver .

## **Arrêt d'un canal et conditions d'échec**

Lorsque MQIPT détecte la fermeture (normale ou anormale) d'un canal IBM MQ, il propage la fermeture du canal. Si vous fermez une route à l'aide de MQIPT, tous les canaux passant par cette route sont fermés.

MQIPT propose une fonction de délai d'inactivité facultative. Si MQIPT détecte qu'un canal a été inactif pendant une période de temps supérieure au délai d'attente, il arrête immédiatement les deux connexions en question.

Les systèmes IBM MQ à chaque extrémité du canal perçoivent ces conditions d'arrêt anormal soit comme des échecs réseau, soit comme un arrêt du canal par leur partenaire. Le canal peut alors redémarrer et effectuer une reprise (si l'échec se produit durant une période d'attente de validation de protocole) comme si MQIPT n'était pas utilisé.

## **Sécurité des messages**

La gestion répartie des files d'attente IBM MQ garantit la distribution des messages. C'est notamment le cas lorsque MQIPT est présent entre les deux extrémités du canal. MQIPT ne stocke pas de données de message et n'est pas impliqué dans la procédure de point de synchronisation qui garantit la distribution des messages.

Lorsque vous utilisez des messages IBM MQ rapides et non persistants, si la route MQIPT échoue ou est redémarrée lorsqu'un message IBM MQ est en cours de transfert, celui-ci peut être perdu. Avant de redémarrer la route, assurez-vous que tous les canaux IBM MQ utilisant la route MQIPT sont inactifs.

Pour plus d'informations sur la sécurité des messages dans IBM MQ, voir [Safety of messages](#).

## **Gestionnaires de files d'attente multi-instances et haute disponibilité**

MQIPT peut être utilisé avec des gestionnaires de files d'attente multi-instances dans des environnements à haute disponibilité.

MQIPT ne dispose d'aucun état permanent et il n'y a donc aucun avantage à basculer MQIPT sur un autre système. Il est préférable de disposer de plusieurs instances de MQIPT avec des fichiers de configuration `mqipt.conf` identiques, s'exécutant sur différents systèmes. Surveillez la disponibilité de chaque instance de MQIPT et redémarrez l'instance si nécessaire (sur le même système). Vous disposez ainsi d'un ensemble d'instances identiques de MQIPT pouvant être utilisées pour acheminer les connexions. Vous devez ensuite vous assurer qu'IBM MQ peut acheminer les connexions à MQIPT et que MQIPT peut les transmettre au gestionnaire de files d'attente de destination.

Les canaux IBM MQ sortants peuvent être dirigés vers une instance MQIPT disponible de plusieurs façons, par exemple :

- Utilisez un équilibreur de charge ou un routeur à haute disponibilité, tel qu'IBM Network Dispatcher du produit WebSphere Edge Components.
- Spécifiez plusieurs noms de connexion dans la définition de canal IBM MQ, sous forme de liste dont les éléments sont séparés par une virgule. IBM MQ tente alors de se connecter à chaque adresse MQIPT tour à tour jusqu'à ce qu'il trouve une instance MQIPT disponible.

Vous devez également diriger les connexions de MQIPT vers le gestionnaire de files d'attente de destination. Si la configuration à haute disponibilité garantit que l'adresse IP est reprise en ligne avec le gestionnaire de files d'attente, aucune configuration MQIPT spéciale n'est requise : indiquez l'adresse IP de destination dans la propriété de route **Destination**. Laissez ensuite l'opération de reprise en ligne déplacer l'adresse IP avec le gestionnaire de files d'attente.

Toutefois, si l'adresse IP du gestionnaire de files d'attente change après une reprise en ligne, vous devez configurer MQIPT pour qu'il transmette la connexion à la destination appropriée. Plusieurs méthodes sont possibles :

- Ecrivez un exit de routage qui vérifie les adresses IP et le numéro de port qui sont accessibles, puis remplacez la destination de routage de chaque connexion. Des exemples d'exit de routage sont fournis avec MQIPT. Ils peuvent être adaptés à cet effet.
- Utilisez un équilibreur de charge à haute disponibilité pour rediriger la connexion.
- Définissez plusieurs routes MQIPT, une pour chaque adresse IP et pour chaque port sur lequel le gestionnaire de files d'attente peut être en cours d'exécution. Dirigez ensuite les connexions IBM MQ vers les différentes routes MQIPT, par exemple en répertoriant tous les numéros de port et toutes les adresses IP de la route dans une liste dont les éléments sont séparés par une virgule, dans le nom de connexion du canal de communications sortantes.

Il est également important d'optimiser tous les composants de bout en bout sur chemin d'accès du réseau :

1. L'échec des tentatives de connexion à des systèmes indisponibles doit être rapide, afin que les tentatives de reconnexion puissent passer à la première destination disponible.

Pour les routes MQIPT SSL, réglez la propriété de route **SSLClientConnectTimeout** afin de garantir un délai court d'échec de connexion aux destinations indisponibles. Consultez la documentation IBM MQ pour plus d'informations sur le réglage du paramètre IBM MQ. De même, consultez la documentation de votre système d'exploitation pour prendre connaissance des détails du réglage de TCP/IP pour le système d'exploitation. Dans tous les cas, les tentatives de connexion ayant échoué doivent renvoyer rapidement un incident réseau (par exemple, un paquet de réinitialisation TCP), ou doivent expirer sans délai inutile.

2. Les connexions actives à un système ayant échoué doivent être éliminées rapidement afin de permettre l'établissement de nouvelles connexions.

Vous devez également prendre en compte l'impact d'une reprise en ligne lorsque les connexions utilisent intensément MQIPT. Il est probable que les connexions réseau soient interrompues pendant une reprise en ligne. Pour des applications client, vous pouvez utiliser la fonction de reconnexion automatique du client d'IBM MQ pour rétablir des connexions interrompues. Pour les canaux de messages, vous pouvez spécifier un intervalle court entre les nouvelles tentatives afin de garantir une reconnexion rapide du canal. Consultez la documentation IBM MQ pour plus d'informations sur la reconnexion automatique du client et la configuration de la relance des canaux de message.



## Remarques

---

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Pour plus de détails, référez-vous aux documents d'annonce disponibles dans votre pays, ou adressez-vous à votre partenaire commercial IBM. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout autre élément fonctionnellement équivalent peut être utilisé, s'il n'enfreint aucun droit d'IBM. 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 EMEA Director of Licensing  
IBM Corporation  
Tour Descartes  
Armonk, NY 10504-1785  
U.S.A.

Pour toute demande d'informations relatives au jeu de caractères codé sur deux octets, contactez le service de propriété intellectuelle IBM ou envoyez vos questions par courrier à l'adresse suivante :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japon

**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 logiciels décrits dans ce document.

Les références à des sites Web non IBM sont fournies à titre d'information uniquement et n'impliquent en aucun cas une adhésion aux données qu'ils contiennent. 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 de WebSphere 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 comme 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 d'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: