

9.4

*Planification pour 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 219.

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

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

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

---

# Table des matières

<b>Planification.....</b>	<b>5</b>
Types d'édition IBM MQ : remarques sur la planification.....	6
Remarques relatives à IBM MQ et IBM MQ Appliance sur site pour la mise en conformité avec le RGPD.....	9
Architectures basées sur un seul gestionnaire de files d'attente.....	19
Architectures basées sur plusieurs gestionnaires de files d'attente.....	20
Planification de vos files d'attente réparties et de vos clusters.....	21
Planification de votre réseau de publication / abonnement distribué.....	78
Planification de vos exigences de stockage et de performances sur Multiplatforms.....	120
Espace disque requis sur Multiplatforms.....	120
Planification de la prise en charge du système de fichiers sur Multiplatforms.....	123
Planification de la prise en charge du système de fichiers pour MFT sur Multiplatforms.....	151
Choix de la journalisation circulaire ou linéaire sur Multiplatforms.....	152
Mémoire partagée sur AIX.....	153
Ressources IPC IBM MQ et UNIX System V.....	153
Priorité des processus IBM MQ et UNIX.....	153
Planning your IBM MQ environment on z/OS.....	154
Planning for your queue manager.....	154
Planning your channel initiator.....	183
Planning your queue sharing group (QSG).....	187
Planning for backup and recovery.....	200
Planning your z/OS UNIX environment.....	209
Planning for Advanced Message Security.....	209
Planning for Managed File Transfer.....	210
Planning to use the IBM MQ Console and REST API on z/OS .....	216
<b>Remarques.....</b>	<b>219</b>
Documentation sur l'interface de programmation.....	220
Marques.....	220



# Planification d'une architecture IBM MQ

---


Lors de la planification de l'environnement IBM MQ, prenez en compte la prise en charge fournie par IBM MQ pour les architectures à un ou plusieurs gestionnaires de files d'attente et pour les styles de messagerie de publication/abonnement et point à point. Prévoyez également vos exigences en matières de ressources et votre utilisation des fonctions de consignation et de sauvegarde.

## Pourquoi et quand exécuter cette tâche

Avant de planifier votre architecture IBM MQ, familiarisez-vous avec les concepts de base de IBM MQ. Voir [IBM MQ Technical overview](#).

Les architectures IBM MQ vont d'architectures simples utilisant un seul gestionnaire de files d'attente à des réseaux plus complexes de gestionnaires de files d'attente interconnectés. Plusieurs gestionnaires de files d'attente sont connectés ensemble à l'aide de techniques de mise en file d'attente répartie. Pour plus d'informations sur la planification d'architectures à gestionnaire de files d'attente unique et à gestionnaires de files d'attente multiples, voir les rubriques suivantes:

- [«Architectures basées sur un seul gestionnaire de files d'attente»](#), à la page 19
- [«Architectures basées sur plusieurs gestionnaires de files d'attente»](#), à la page 20
  - [«Planification de vos files d'attente réparties et de vos clusters»](#), à la page 21
  - [«Planification de votre réseau de publication / abonnement distribué»](#), à la page 78

 Sous IBM MQ for z/OS, vous pouvez utiliser des files d'attente partagées et des groupes de partage de files d'attente pour vous permettre d'implémenter l'équilibrage de charge et vos applications IBM MQ pour qu'elles soient évolutives et à haute disponibilité. Pour plus d'informations sur les files d'attente partagées et les groupes de partage de files d'attente, voir [Files d'attente partagées et groupes de partage de files d'attente](#).

IBM MQ fournit deux modèles d'édition différents:

- L'édition Long Term Support (LTS) convient parfaitement aux systèmes nécessitant un déploiement à long terme et une stabilité maximale.
- L'édition Continuous Delivery (CD) est destinée aux systèmes qui doivent exploiter rapidement les dernières améliorations fonctionnelles d' IBM MQ.

Les deux types d'édition sont installés de la même manière, mais il existe des considérations relatives à la prise en charge et à la migration que vous devez comprendre. Pour plus d'informations, voir [Types d'édition et gestion des versions IBM MQ](#).

Pour plus d'informations sur la planification de plusieurs installations, sur les exigences de stockage et de performances et sur l'utilisation des clients, voir les autres sous-rubriques.

## Concepts associés

[Types d'édition et gestion des versions IBM MQ](#)

[«Planning your IBM MQ environment on z/OS»](#), à la page 154

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

[Disponibilité, reprise et redémarrage](#)

## Tâches associées

[Vérification de la configuration](#)

[S'assurer que les messages ne sont pas perdus \(consignation\)](#)

## Types d'édition IBM MQ : remarques sur la planification

Les deux principaux types d'édition pour IBM MQ sont Long Term Support (LTS) et Continuous Delivery (CD). Pour chaque plateforme prise en charge, le type d'édition que vous choisissez affecte la commande, l'installation, la maintenance et la migration.

Pour des informations détaillées sur les types d'édition, voir [Types d'édition et gestion des versions d'IBM MQ](#).

### Considérations relatives à IBM MQ for Multiplatforms



#### Ordre

Dans Passport Advantage, il existe deux eAssemblies distincts pour IBM MQ 9.4. L'une contient des images d'installation pour l'édition IBM MQ 9.4.0 Long Term Support et l'autre contient des images d'installation pour l'édition IBM MQ 9.4.x Continuous Delivery. Téléchargez les images d'installation à partir de l'eAssembly en fonction de l'édition de votre choix.

Toutes les versions de IBM MQ, et pour IBM MQ 9.4 les éditions LTS et CD, appartiennent au même ID produit.

Le droit d'utiliser IBM MQ s'étend à l'ensemble du produit (PID), sous réserve des contraintes des composants sous licence et des métriques de tarification. Cela signifie que vous pouvez choisir librement entre les images d'installation d'édition LTS et CD pour IBM MQ 9.4.

#### Installation

Après avoir téléchargé une image d'installation à partir de Passport Advantage, vous devez sélectionner pour l'installation uniquement les composants pour lesquels vous avez acquis une autorisation d'utilisation. Voir [Informations sur la licence IBM MQ](#) pour plus d'informations sur les composants installables inclus pour chaque composant facturable.

Vous pouvez installer les éditions IBM MQ 9.4.0 LTS et IBM MQ 9.4.x CD sur la même image de système d'exploitation. Dans ce cas, les composants apparaissent comme des installations distinctes prises en charge par la prise en charge multiversion d'IBM MQ. Chaque version comporte des ensembles distincts de gestionnaires de files d'attente associés à cette version.

Chaque nouvelle édition de CD est fournie en tant qu'image d'installation. La nouvelle édition de CD peut être installée avec une édition existante ou une édition antérieure de CD peut être mise à jour par le programme d'installation vers la nouvelle édition.

Les éditions CD contiennent des améliorations fonctionnelles ainsi que l'ensemble le plus récent de correctifs et de mises à jour de sécurité. Chaque édition d'CD est cumulative et remplace intégralement toutes les éditions précédentes de cette version d'IBM MQ. Par conséquent, vous pouvez ignorer une édition CD spécifique si elle ne contient aucune fonction pertinente pour votre entreprise.

#### Maintenance

L'édition LTS est prise en charge par l'application de groupes de correctifs, qui fournissent des correctifs d'incident, et de mises à jour de sécurité cumulatives (CSU), qui fournissent des correctifs de sécurité. Les groupes de correctifs et les CSU sont mis à disposition périodiquement et sont cumulatifs.

Pour CD, les unités de service (CSU) sont produites pour la dernière édition de CD uniquement, qui peut se trouver sur une version ultérieure.

Il peut arriver que l'équipe de support IBM vous demande d'appliquer un correctif temporaire. Les correctifs temporaires sont également appelés correctifs d'urgence ou de test et sont utilisés pour appliquer des mises à jour urgentes qui ne peuvent pas attendre la prochaine livraison de maintenance.

#### Migration entre l'édition LTS et l'édition CD

Il existe des contraintes et des limitations mais, en règle générale, un seul gestionnaire de files d'attente peut être migré à partir du code d'édition LTS vers le code d'édition CD ou à partir du code d'édition CD vers le code d'édition LTS, à condition que l'édition cible soit supérieure à celle utilisée avant la migration.

Deux approches sont possibles:

- Installez la nouvelle édition du code de sorte qu'une installation existante de IBM MQ soit mise à jour. Tous les gestionnaires de files d'attente associés à l'installation utilisent la nouvelle édition du code lorsqu'ils sont démarrés.
- Installez la nouvelle édition de code en tant que nouvelle installation, puis déplacez des instances de gestionnaire de files d'attente individuelles vers la nouvelle installation à l'aide de la commande `setmqm`.

Lorsqu'un gestionnaire de files d'attente démarre en exécutant une édition de code CD, le niveau de commande du gestionnaire de files d'attente est mis à jour pour indiquer le nouveau niveau d'édition. Cela signifie que les nouvelles fonctions fournies dans l'édition sont activées et que vous ne pouvez plus redémarrer le gestionnaire de files d'attente à l'aide d'une édition de code avec un nombre VRM inférieur.

## Considérations relatives à IBM MQ for z/OS



### Ordre

Lors de la commande de IBM MQ for z/OS 9.4, deux fonctions distinctes sont proposées sur ShopZ. Les fonctions correspondent à l'édition LTS et à l'édition CD. Les deux fonctions sont applicables au même ID de produit (PID). Il s'agit de l'ID de produit sous licence. Par conséquent, lorsqu'une fonction est sous licence, il est autorisé à utiliser la fonction alternative si nécessaire. Lors de la commande, vous sélectionnez le dispositif correspondant à l'édition LTS ou à l'édition CD.

Si vous sélectionnez des produits à inclure dans un ServerPac, vous ne pouvez pas choisir à la fois l'édition LTS et l'édition CD dans la même commande ServerPac, car les produits ne peuvent pas être installés par SMP/E dans la même zone cible.

### Installation

Les éditions LTS et CD sont fournies dans des ensembles distincts de FMID. Notez que ces FMID ne peuvent pas être installés dans la même zone cible SMP/E. Si vous avez besoin des éditions LTS et CD :

- Installez l'édition LTS et l'édition CD dans des zones cible distinctes.
- Gérez des bibliothèques cible et de distribution distinctes pour les deux éditions.

Si votre gestionnaire de files d'attente se trouve dans un groupe de partage de files d'attente, lorsque vous effectuez une mise à niveau vers la dernière version du CD, vous devez mettre à niveau tous les gestionnaires de files d'attente du groupe.

Le niveau de commande d'un gestionnaire de files d'attente correspond au niveau VRM à trois chiffres. Un programme IBM MQ peut appeler MQINQ, en transmettant le sélecteur MQIA\_COMMAND\_LEVEL, pour obtenir le niveau de commande du gestionnaire de files d'attente auquel il est connecté.

Etant donné que les éditions utilisent des FMID différents, vous ne pouvez pas mettre à jour une édition CD avec maintenance pour une édition LTS ou l'inverse. De même, il n'est pas possible de changer une version du code produit d'une édition LTS à une édition CD ou inversement. Toutefois, vous pouvez changer de gestionnaire de files d'attente entre les modèles d'édition. Voir [Migration entre l'édition LTS et l'édition CD](#).

### Remarque :

Les éditions IBM MQ 9.0.x et IBM MQ 9.1.x CD avaient des FMID dépendants de la version et de l'édition distincts. Par conséquent, le passage de 9.0.x CD à 9.1.x CD a nécessité au moins une installation SMP/E complète.

Depuis la IBM MQ for z/OS 9.2.0, l'édition CD utilise un ensemble de FMID qui restent les mêmes pour toutes les éditions IBM MQ for z/OS avec le numéro de version 9. Etant donné que chaque nouvelle version de IBM MQ est disponible à la fois en tant qu'édition CD et en tant qu'édition LTS, vous pouvez mettre à niveau les éditions CD en appliquant des PTF à une seule installation SMP/E, même lorsque vous franchissez une limite de version majeure. Par exemple, vous pouvez passer de IBM MQ for z/OS 9.2.0 CD,

à IBM MQ for z/OS 9.2.2 CD, à IBM MQ for z/OS 9.2.4 CD, à IBM MQ for z/OS 9.3.0 CD, simplement en appliquant des PTF.

Vous pouvez faire la distinction entre une édition LTS et CD avec le même niveau de régulateur de tension en examinant le message `CSQY000I` dans le journal des travaux du gestionnaire de files d'attente.

## Maintenance

IBM MQ for z/OS utilise des PTF pour la maintenance.

**LTS** Les PTF sont spécifiques à un ensemble particulier de bibliothèques correspondant au niveau d'édition spécifique. Pour les fonctions UNIX System Services (c'est-à-dire JMS et WEB UI, Connector Pack et Managed File Transfer), les PTF z/OS sont alignées directement avec les groupes de correctifs Multiplatforms et les mises à jour de sécurité cumulatives (CSU). Ces correctifs sont cumulatifs et sont disponibles en même temps que le groupe de correctifs Multiplatforms ou CSU équivalent.

**CD** Les unités de contrôle CD ne sont généralement pas disponibles entre les éditions CD, mais sont incluses dans la prochaine édition de IBM MQ for z/OS CD . Vous pouvez également contacter le support pour demander un ++ USERMOD.

Les autres correctifs sur IBM MQ for z/OS sont des correctifs distincts sur des parties particulières. Ces correctifs résolvent des problèmes spécifiques, ne sont pas cumulatifs et sont mis à disposition au fur et à mesure de leur production.

## Migration entre l'édition LTS et l'édition CD

Il existe des contraintes et des limitations, mais en règle générale, un seul gestionnaire de files d'attente peut être migré à partir du code d'édition LTS vers le code d'édition CD ou à partir du code d'édition CD vers le code d'édition LTS , à condition que l'édition cible soit supérieure à celle utilisée avant la migration.

Depuis la IBM MQ for z/OS 9.2.0, vous pouvez effectuer une migration entre les éditions CD et LTS avec le même gestionnaire de ressources virtuel autant de fois que nécessaire, et sans impact sur la possibilité de rétromigration. Par exemple, un gestionnaire de files d'attente peut être démarré dans IBM MQ for z/OS 9.3.0 LTS, puis arrêté et démarré dans IBM MQ for z/OS 9.3.0 CD, puis arrêté et démarré dans IBM MQ for z/OS 9.3.0 LTS.

IBM MQ for z/OS a traditionnellement fourni une fonction de rétromigration (rétromigration) pour qu'après une période d'exécution suivant une migration, vous puissiez revenir à l'édition précédente. Cette fonction est conservée pour les éditions LTS et pour les éditions CD avec un modificateur 0 tel que 9.3.0 CD, mais elle n'est pas possible lorsque la source ou la cible d'une migration est une édition CD avec un numéro de modificateur différent de zéro, par exemple 9.2.5 ou 9.3.1.

Voici des scénarios de migration valides qui illustrent le fonctionnement de ce principe:

Edition source	Edition de destination	Remarques
9.1.0 LTS	9.4.0 LTS ou 9.4.0 CD	La rétromigration n'est pas prise en charge car 9.1.0 LTS n'est pas pris en charge de manière standard.
9.2.0 LTS	9.4.0 LTS ou 9.4.0 CD	Rétromigration prise en charge.
9.3.0 LTS	9.4.0 LTS ou 9.4.0 CD	Rétromigration prise en charge.
CD 9.3.5	9.4.0 LTS ou 9.4.0 CD	La rétromigration n'est pas prise en charge car l'édition source est CD avec un modificateur différent de zéro.



Édition source	Édition de destination	Remarques
9.4.0 LTS ou 9.4.0 CD	CD 9.4.1	La rémigration n'est pas prise en charge car l'édition de destination est CD avec un modificateur différent de zéro.  Write to operator with reply CSQY041D est émis pour confirmer la migration.

#### Tâches associées

 Application et retrait de maintenance sous z/OS

#### Information associée

[Téléchargement en cours IBM MQ 9.4](#)

## Remarques relatives à IBM MQ et IBM MQ Appliance sur site pour la mise en conformité avec le RGPD

### Pour les ID produit :

#### Distribué

- IBM MQ/IBM MQ Advanced - 5724-H72
- IBM MQ for HPE NonStop - 5724-A39

#### z/OS

- IBM MQ for z/OS - 5655-MQ9
- IBM MQ for z/OS Value Unit Edition - 5655-VU9
- IBM MQ Advanced for z/OS - 5655-AV9
- IBM MQ Advanced for z/OS Value Unit Edition - 5655-AV1

#### IBM MQ Appliance

- IBM MQ Appliance M2003 - 5900-ALJ
- IBM MQ Appliance M2002 - 5737-H47

### Notification :

Servez-vous de ce document pour préparer votre mise en conformité avec le RGPD. Il contient des informations sur les fonctions d'IBM MQ que vous pouvez configurer, ainsi qu'aux aspects de l'utilisation du produit qu'il vous faut prendre en compte pour mieux organiser votre mise en conformité avec le RGPD. Ces informations ne sont pas exhaustives, dans la mesure où les clients peuvent choisir de configurer les fonctions de nombreuses manières différentes et où le produit peut être lui-même utilisé bien différemment d'un client à un autre et avec des applications et systèmes tiers.

**Il appartient aux clients de s'assurer de leur conformité avec les différentes lois en vigueur, y compris le Règlement Général sur le Protection des Données de l'Union Européenne. Il relève de la seule responsabilité du client de consulter les services juridiques compétents aussi bien pour identifier et interpréter les lois et règlements susceptibles d'affecter son activité, que pour toute action à entreprendre pour se mettre en conformité avec ces lois et réglementations.**

**Les produits, services et autres fonctionnalités décrites dans ce document ne sont pas valables pour toutes les situations du client, et leur disponibilité peut être limitée. IBM ne fournit ni audit ni conseil juridique, ni déclaration, ni garantie que ses services ou produits assurent au client d'être en conformité avec la loi.**

## Table des matières

1. [RGPD](#)
2. [Configuration du produit pour le règlement RGPD](#)
3. [Cycle de vie des données](#)
4. [Collecte de données](#)
5. [Stockage des données](#)
6. [Accès aux données](#)
7. [Traitement des données](#)
8. [Suppression des données](#)
9. [Surveillance des données](#)
10. [Capacité de restreindre l'utilisation des données personnelles](#)
11. [Gestion des fichiers](#)

## RGPD

Le Règlement général sur la protection des données (RGPD) a été adopté par l'Union européenne et s'applique à partir du 25 mai 2018.

### Pourquoi le RGPD est-il important ?

Le RGPD établit un cadre réglementaire plus solide en matière de protection de données pour le traitement des données à caractère personnel des personnes concernées. Il apporte :

- Confère des droits nouveaux et améliorés aux personnes concernées
- Elargit la définition des données à caractère personnel
- Stipule de nouvelles obligations pour les responsables du traitement
- Prévoit d'éventuelles sanctions financières significatives en cas de non-conformité
- Impose l'obligation de notifier les violations de données

### Pour en savoir plus sur le RGPD :

- [Portail d'information RGPD de l'Union européenne](#)
- [Site web ibm.com/GDPR](http://ibm.com/GDPR)

## Configuration du produit : remarques relatives à la mise en conformité avec le RGPD

Les sections qui suivent décrivent la configuration de IBM MQ pour aider votre organisation à se préparer à la mise en oeuvre du RGPD.

### Cycle de vie des données

IBM MQ est un produit middleware orienté message transactionnel qui permet aux applications d'échanger de manière asynchrone les données fournies par les applications. IBM MQ prend en charge une gamme d'API de messagerie, de protocoles et de ponts dans le but de connecter des applications. En tant que tel, IBM MQ peut être utilisé pour échanger de nombreuses formes de données, dont certaines pourraient potentiellement être soumises au RGPD. Il existe plusieurs produits tiers avec lesquels IBM MQ peut échanger des données. Certains d'entre eux appartiennent à IBM, mais de nombreux autres sont fournis par d'autres fournisseurs de technologie. Le [site Web des rapports de compatibilité des produits logiciels](#) fournit la liste des logiciels associés. Pour toute information concernant la mise en conformité d'un produit tiers avec le RGPD, consultez la documentation de ce produit. Les administrateurs IBM MQ contrôlent la manière dont IBM MQ interagit avec les données qui le transitent, par la définition des files d'attente, des rubriques et des abonnements.

### Quels types de données transitent par IBM MQ?

Comme IBM MQ fournit un service de messagerie asynchrone pour les données d'application, il n'existe pas de réponse définitive à cette question car les cas d'utilisation varient en fonction du déploiement de l'application. Les données de message d'application sont conservées dans les fichiers de file d'attente (ensembles de pages ou unité de couplage sous z/OS), les journaux et les archives et le message peut lui-même contenir des données régies par le RGPD. Les données de message fournies par l'application peuvent également être incluses dans les fichiers collectés à des fins d'identification des problèmes, tels que les journaux d'erreurs, les fichiers de trace et les fichiers FFST. Dans l'application z/OS, les données de message fournies peuvent également être incluses dans les vidages d'espace adresse ou d'unité de couplage.

Voici quelques exemples typiques de données personnelles qui peuvent être échangées à l'aide de IBM MQ:

- Les employés du client (par exemple, IBM MQ peut être utilisé pour connecter les systèmes de paie ou de ressources humaines du client)
- Les données personnelles des clients du client (par exemple, IBM MQ peut être utilisé par un client pour échanger des données entre des applications liées à ses clients, telles que la prise de prospects et le stockage de données dans leur système CRM).
- Les données personnelles sensibles des clients du client (par exemple, IBM MQ peut être utilisé dans des contextes métier nécessitant l'échange de données personnelles, tels que des enregistrements de soins de santé HL7-based lors de l'intégration d'applications cliniques).

Outre les données de message fournies par l'application, IBM MQ traite les types de données suivants:

- Données d'authentification (telles que le nom d'utilisateur et les mots de passe, les clés d'API, etc.)
- Les informations personnelles d'identification techniques (comme les ID d'unité, les identificateurs reposant sur l'utilisation, les adresses IP, etc., lorsqu'ils sont liés à un individu)

### **Données personnelles utilisées pour les contacts en ligne avec IBM**

Les clients IBM MQ peuvent soumettre des commentaires / feedback/demandes en ligne pour contacter IBM sur des sujets IBM MQ de différentes manières, principalement:

- La zone de commentaires publics sur les pages de la zone [IBM MQ sur IBM Developer](#)
- Zone de commentaires publics sur les pages des informations produit [IBM MQ dans IBM Documentation](#)
- Commentaires publics dans les [forums de support IBM](#)
- Commentaires publics dans [IBM Integration Ideas](#)

En général, seuls le nom et l'adresse e-mail du client sont utilisés, afin de permettre des réponses personnelles pour l'objet du contact, et l'utilisation des données à caractère personnel se conforme à la [Déclaration IBM de confidentialité sur Internet](#).

### **Collecte de données**

IBM MQ peut être utilisé pour collecter des données personnelles. Lors de l'évaluation de votre utilisation de IBM MQ et de vos besoins pour répondre aux exigences du RGPD, vous devez prendre en compte les types de données à caractère personnel qui, dans votre cas, transitent par IBM MQ. Vous souhaitez peut-être prendre en considération des aspects tels que:

- Comment les données arrivent-elles dans vos gestionnaires de files d'attente? (A travers quels protocoles? Les données sont-elles chiffrées? Les données sont-elles signées?)
- Comment les données sont-elles envoyées à partir de vos gestionnaires de files d'attente? (A travers quels protocoles? Les données sont-elles chiffrées? Les données sont-elles signées?)
- Comment les données sont-elles stockées lorsqu'elles passent par un gestionnaire de files d'attente? (Toute application de messagerie peut écrire des données de message sur un support avec état, même si un message n'est pas persistant. Êtes-vous conscient de la manière dont les fonctions de messagerie pourraient potentiellement exposer des aspects des données de message d'application qui transitent par le produit?)

- Comment les données d'identification sont-elles collectées et stockées là où IBM MQ en a besoin pour accéder aux applications tierces?

IBM MQ peut avoir besoin de communiquer avec d'autres systèmes et services qui requièrent une authentification, par exemple LDAP. Si nécessaire, les données d'authentification (ID utilisateur, mots de passe) sont configurées et stockées par IBM MQ pour être utilisées dans de telles communications. Dans la mesure du possible, évitez d'utiliser des données d'identification personnelles pour l'authentification IBM MQ. Prenez en compte la protection du stockage utilisé pour les données d'authentification. (Voir Stockage de données ci-dessous.)

## Stockage de Données

Lorsque les données de message transitent par les gestionnaires de files d'attente, IBM MQ conserve (peut-être plusieurs copies) ces données directement sur le support avec état. Les utilisateurs IBM MQ peuvent envisager de sécuriser les données de message lorsqu'elles sont au repos.

Les éléments suivants mettent en évidence les domaines dans lesquels IBM MQ conserve les données fournies par l'application, que les utilisateurs peuvent souhaiter prendre en compte lors de la vérification de la conformité au RGPD.

- Files d'attente de messages d'application:

IBM MQ fournit des files d'attente de messages pour permettre l'échange de données asynchrone entre les applications. Les messages non persistants et persistants stockés dans une file d'attente sont écrits sur un support avec état.

- Files d'attente de l'agent de transfert de fichiers:

IBM MQ Managed File Transfer utilise des files d'attente de messages pour coordonner le transfert fiable des données de fichier, les fichiers contenant des données personnelles et les enregistrements des transferts sont stockés dans ces files d'attente.

- Files d'attente de transmission:

Pour transférer les messages de manière fiable entre les gestionnaires de files d'attente, les messages sont stockés temporairement dans les files d'attente de transmission.

- Files d'attente des messages non livrés:

Dans certains cas, les messages ne peuvent pas être placés dans une file d'attente de destination et sont stockés dans une file d'attente de rebut, si une telle file d'attente est configurée sur le gestionnaire de files d'attente.

- Files d'attente d'annulation:

Les interfaces de messagerie JMS et XMS fournissent une fonction qui permet de déplacer des messages incohérents vers une file d'attente d'annulation après un certain nombre d'annulations pour permettre le traitement d'autres messages valides.

- File d'attente d'erreurs AMS:

IBM MQ Advanced Message Security déplace les messages qui ne sont pas conformes à une règle de sécurité vers SYSTEM.PROTECTION.ERROR.QUEUE est similaire à la file d'attente des messages non livrés.

- Publications conservées:

IBM MQ fournit une fonction de publication conservée pour permettre aux applications d'abonnement de rappeler une publication précédente.

- Livraison différée:

IBM MQ prend en charge la fonction de délai de distribution JMS 2.0 et Jakarta Messaging 3.0 qui permet aux messages d'être distribués ultérieurement à leur destination. Les messages qui n'ont pas encore été distribués sont stockés dans SYSTEM.DDELAY.LOCAL.QUEUE.

Pour en savoir plus :

- [Journalisation: S'assurer que les messages ne sont pas perdus](#)

- [Paramètres de file d'attente de l'agent MFT](#)
- [Utilisation de la file d'attente de rebut](#)
- [Gestion des messages incohérents dans IBM MQ classes for JMS](#)
- [Gestion des erreurs AMS](#)
- [Publications conservées](#)
- [JMS 2.0 délai de distribution](#)

Les éléments suivants mettent en évidence les domaines dans lesquels IBM MQ peut indirectement conserver des données d'application fournies que les utilisateurs peuvent également souhaiter prendre en compte lors de la mise en conformité avec le RGPD.

- **Messagerie de routage de trace:**

IBM MQ fournit des fonctions d'acheminement de trace, qui enregistrent l'acheminement d'un message entre les applications. Les messages d'événement générés peuvent inclure des informations personnelles techniquement identifiables telles que des adresses IP.

- **Trace de l'activité de l'application:**

IBM MQ fournit une trace d'activité d'application, qui enregistre les activités d'API de messagerie des applications et des canaux, la trace d'activité d'application peut enregistrer le contenu des données de message fournies par l'application dans les messages d'événement.

- **Trace de service:**

IBM MQ fournit des fonctions de trace de service, qui enregistrent les chemins de code interne par lesquels transitent les données de message. Dans le cadre de ces fonctions, IBM MQ peut enregistrer le contenu des données de message fournies par l'application dans des fichiers de trace stockés sur le disque.

- **Événements du gestionnaire de files d'attente:**

IBM MQ peut générer des messages d'événement pouvant inclure des données personnelles, telles que des événements d'autorité, de commande et de configuration.

Pour en savoir plus :

- [Messagerie de routage de trace](#)
- [Utilisation de la trace](#)
- [Surveillance d'événements](#)
- [Événements du gestionnaire de files d'attente](#)

Pour protéger l'accès aux copies des données de message fournies par l'application, prenez en compte les actions suivantes:

- Limitez l'accès des utilisateurs privilégiés aux données IBM MQ dans le système de fichiers, par exemple en limitant l'appartenance des utilisateurs au groupe'mqm'sur les plateformes UNIX and Linux® .
- Limitez l'accès des applications aux données IBM MQ via des files d'attente dédiées et le contrôle d'accès. Le cas échéant, évitez le partage inutile des ressources telles que les files d'attente entre les applications et fournissez un contrôle d'accès granulaire aux ressources de file d'attente et de rubrique.
- Limitez l'accès aux copies répliquées des données IBM MQ dans les configurations à haute disponibilité ou de reprise après incident et sécurisez les connexions utilisées pour la réplique.
- Utilisez IBM MQ Advanced Message Security pour fournir la signature et / ou le chiffrement de bout en bout des données de message.
- Utilisez le chiffrement de niveau fichier ou volume pour protéger les répertoires ou les systèmes de fichiers pouvant contenir des données, des traces ou des journaux IBM MQ .
- Après avoir téléchargé la trace de service dans IBM, vous pouvez supprimer vos fichiers de trace de service et vos données FFST si vous vous souciez du contenu contenant potentiellement des données personnelles.

Pour en savoir plus :

- [Utilisateurs privilégiés](#)
- [Planification de la prise en charge du système de fichiers sur Multiplatforms](#)
- [Chiffrement du système de fichiers sur le IBM MQ Appliance](#)

Un administrateur IBM MQ peut configurer un gestionnaire de files d'attente avec des données d'identification (nom d'utilisateur et mot de passe, clés d'API, etc.) pour des services tiers tels que LDAP. Ces données sont généralement stockées dans le répertoire de données du gestionnaire de files d'attente protégé par les droits du système de fichiers.

Lorsqu'un gestionnaire de files d'attente IBM MQ est créé, le répertoire de données est configuré avec un contrôle d'accès de groupe de sorte que IBM MQ puisse lire les fichiers de configuration et utiliser les données d'identification pour se connecter à ces systèmes. Les administrateurs IBM MQ sont considérés comme des utilisateurs privilégiés et sont membres de ce groupe. Ils disposent donc d'un accès en lecture aux fichiers. Certains fichiers sont brouillés mais ne sont pas chiffrés. Pour cette raison, afin de protéger complètement l'accès aux données d'identification, vous devez prendre en compte les actions suivantes:

- Limitez l'accès des utilisateurs privilégiés aux données IBM MQ , par exemple en limitant l'appartenance du groupe'mqm'sur les plateformes UNIX and Linux .
- Utilisez le chiffrement au niveau du fichier ou du volume pour protéger le contenu du répertoire de données du gestionnaire de files d'attente.
- Chiffrez les sauvegardes du répertoire de configuration de production et stockez-les avec les contrôles d'accès appropriés.
- Envisagez de fournir des traces d'audit pour les échecs d'authentification, le contrôle d'accès et les modifications de configuration avec des événements de sécurité, de commande et de configuration.


Pour en savoir plus :

- [Sécurisation d' IBM MQ](#)

## Accès aux données

Les données du gestionnaire de files d'attente IBM MQ sont accessibles via les interfaces de produit suivantes, dont certaines sont conçues pour l'accès via une connexion distante et d'autres pour l'accès via une connexion locale.

- Console IBM MQ [ à distance uniquement ]
- IBM MQ API REST d'administration [ à distance uniquement ]
- IBM MQ API REST de messagerie [ à distance uniquement ]
- MQI [ Local et distant ]
- JMS [ Local et distant ]
- XMS [ Local et distant ]
- IBM MQ Telemetry (MQTT) [ à distance uniquement ]
- IBM MQ Light (AMQP) [ A distance uniquement ]
- Passerelle IBM MQ IMS [ Local uniquement ]
- Pont IBM MQ CICS [ Local uniquement ]
- IBM MQ ponts de protocole MFT [ uniquement distants ]
- IBM MQ ponts Connect:Direct [ à distance uniquement ]
- IBM MQ MQAI [ Local et distant ]
- IBM MQ Commandes PCF [ Local et Remote ]
- IBM MQ Commandes MQSC [ Local et distant ]
- IBM MQ Explorer [ Local et distant ]

- Exits utilisateur IBM MQ [ uniquement locaux ]
- IBM MQ Internet Pass-Thru [ A distance uniquement ]
- Métriques Red Hat® OpenShift® Monitoring (Prometheus) (les métriques sont des données numériques sur les statistiques du gestionnaire de files d'attente)
- IBM MQ Appliance Serial Console [ Local uniquement ]
- IBM MQ Appliance SSH [ à distance uniquement ]
- IBM MQ Appliance API REST [ Uniquement à distance ]
- IBM MQ Appliance Web UI [ Uniquement à distance ]
-  Connecteurs IBM MQ Kafka (Kafka Connect) [ Local et distant ]

Les interfaces sont conçues pour permettre aux utilisateurs d'apporter des modifications à un gestionnaire de files d'attente IBM MQ et aux messages qu'il contient. Les opérations d'administration et de messagerie sont sécurisées de sorte qu'il y a trois étapes à franchir lorsqu'une demande est effectuée ;

- Authentification
- Mappage de rôle
- Autorisation

#### **Authentification:**

Si le message ou l'opération d'administration a été demandé à partir d'une connexion locale, la source de cette connexion est un processus en cours d'exécution sur le même système. L'utilisateur exécutant le processus doit avoir passé toutes les étapes d'authentification fournies par le système d'exploitation. Le nom d'utilisateur du propriétaire du processus à partir duquel la connexion a été établie est vérifié en tant qu'identité. Par exemple, il peut s'agir du nom de l'utilisateur exécutant le shell à partir duquel une application a été démarrée. Les formes d'authentification possibles pour les connexions locales sont les suivantes:

1. Nom d'utilisateur vérifié (système d'exploitation local)
2. Nom d'utilisateur et mot de passe facultatifs (système d'exploitation, LDAP ou 3rd personnalisés)
3. Jeton de sécurité (JWT) IBM MQ uniquement

Si l'action d'administration a été demandée à partir d'une connexion distante, les communications avec IBM MQ sont établies via une interface réseau. Les formes d'identité suivantes peuvent être présentées pour l'authentification via des connexions réseau:

1. Nom d'utilisateur vérifié (à partir du système d'exploitation distant)
2. Nom d'utilisateur et mot de passe (système d'exploitation, LDAP ou 3rd personnalisés)
3. Adresse réseau source (telle que l'adresse IP)
4. X.509 Certificat numérique (authentification SSL/TLS mutuelle)
5. Jetons de sécurité (tels que le jeton LTPA2 ou le jeton JWT)
6. Autre sécurité personnalisée (fonction fournie par les exits 3rd )
7. Clés SSH

L'intégration d' IBM MQ à IBM Cloud Pak for Integration ajoute un nouveau type d'authentification pour IBM MQ Console: Single Sign-On avec Cloud Pak. (CP4I uniquement)

#### **Mappage de rôle:**

Lors de l'étape de mappage de rôle, les données d'identification fournies lors de l'étape d'authentification peuvent être mappées à un autre identificateur utilisateur. A condition que l'ID utilisateur mappé soit autorisé à continuer (par exemple, les administrateurs peuvent être bloqués par des règles d'authentification de canal), puis l'ID utilisateur mappé est reporté à l'étape finale lors de l'autorisation des activités sur les ressources IBM MQ .

#### **Autorisation :**

IBM MQ permet à différents utilisateurs d'avoir des droits différents sur des ressources de messagerie différentes, telles que des files d'attente, des rubriques et d'autres objets de gestionnaire de files d'attente.

### **Activité de journalisation:**

Certains utilisateurs de IBM MQ peuvent avoir besoin de créer un enregistrement d'audit d'accès aux ressources MQ . Des exemples de journaux d'audit souhaitables peuvent inclure des modifications de configuration qui contiennent des informations sur la modification en plus de la personne qui en a fait la demande.

Les sources d'information suivantes sont disponibles pour mettre en oeuvre cette exigence:

1. Un gestionnaire de files d'attente IBM MQ peut être configuré pour générer des événements de commande lorsqu'une commande admin a été exécutée avec succès.
2. Un gestionnaire de files d'attente IBM MQ peut être configuré pour générer des événements de configuration lorsqu'une ressource de gestionnaire de files d'attente est créée, modifiée ou supprimée.
3. Un gestionnaire de files d'attente IBM MQ peut être configuré pour générer un événement d'autorité lorsqu'une vérification d'autorisation échoue pour une ressource.
4. Les messages d'erreur indiquant que les vérifications d'autorisation ont échoué sont consignés dans les journaux des erreurs du gestionnaire de files d'attente.
5. La console IBM MQ écrit des messages d'audit dans ses journaux lorsque l'authentification, les vérifications d'autorisation échouent ou lorsque des gestionnaires de files d'attente sont créés, démarrés, arrêtés ou supprimés.
6. IBM MQ Appliance écrit des messages d'audit dans ses journaux pour enregistrer les connexions utilisateur et les modifications du système.

Lors de la prise en compte de ces types de solutions, les utilisateurs de IBM MQ souhaiteront peut-être prendre en compte les points suivants:

- Les messages d'événement sont non persistants. Par conséquent, lorsqu'un gestionnaire de files d'attente redémarre, les informations sont perdues. Les moniteurs d'événements doivent être configurés pour consommer en permanence tous les messages disponibles et transférer le contenu sur des supports persistants.
- Les utilisateurs privilégiés IBM MQ disposent de privilèges suffisants pour désactiver des événements, effacer des journaux ou supprimer des gestionnaires de files d'attente.

Pour plus d'informations sur la sécurisation de l'accès aux données IBM MQ et la fourniture d'une trace d'audit, voir les rubriques suivantes:

- [IBM MQ](#)
- [Événements de configuration](#)
- [Événements Commande](#)
- [Utilisation des journaux d'erreurs](#)

## **Traitement des Données**

### **Chiffrement à l'aide d'une infrastructure à clé publique:**

Vous pouvez sécuriser les connexions réseau à IBM MQ en spécifiant que les connexions utilisent TLS, qui peut également fournir une authentification mutuelle du côté initiateur de la connexion.

L'utilisation des fonctions de sécurité PKI fournies par les mécanismes de transport constitue la première étape de la sécurisation du traitement des données avec IBM MQ. Cependant, sans activer d'autres fonctions de sécurité, le comportement d'une application consommatrice consiste à traiter tous les messages qui lui sont distribués sans valider l'origine du message ou s'il a été modifié pendant son transit.

Les utilisateurs de IBM MQ qui sont autorisés à utiliser les fonctions Advanced Message Security (AMS) peuvent contrôler la manière dont les applications traitent les données personnelles contenues dans les



messages, via la définition et la configuration des règles de sécurité. Les règles de sécurité permettent d'appliquer la signature numérique et / ou le chiffrement aux données de message entre les applications.

Il est possible d'utiliser des règles de sécurité pour exiger et valider une signature numérique lors de la consommation de messages afin de s'assurer que les messages sont authentiques. Le chiffrement AMS fournit une méthode par laquelle les données de message sont converties d'une forme lisible en une version codée qui ne peut être décodée par une autre application que si elle est le destinataire prévu ou le message et qu'elle a accès à la clé de déchiffrement correcte.

Pour plus d'informations sur l'utilisation de SSL et de certificats pour sécuriser vos connexions réseau, voir les rubriques suivantes dans la documentation du produit IBM MQ :

- [Configuration de la sécurité TLS pour IBM MQ](#)
- [Présentation d'AMS](#)

## Suppression des données

IBM MQ fournit des commandes et des actions d'interface utilisateur pour supprimer les données qui ont été fournies au produit. Cela signifie que les utilisateurs de IBM MQ peuvent supprimer des données relatives à des personnes particulières, si nécessaire.

- Domaines du comportement de IBM MQ à prendre en compte pour la conformité à la suppression des données client RGPD
  - Supprimez les données de message stockées dans une file d'attente d'application en:
    - Suppression de messages individuels à l'aide de l'API de messagerie ou des outils ou à l'aide de l'expiration des messages.
    - En spécifiant que les messages sont non persistants, conservés dans une file d'attente où la classe de message non persistant est normale, puis en redémarrant le gestionnaire de files d'attente.
    - Effacement administratif de la file d'attente.
    - Suppression de la file d'attente.
  - Supprimer les données de publication conservées stockées sur une rubrique par:
    - Spécification de la non-persistance des messages et redémarrage du gestionnaire de files d'attente.
    - Remplacement des données conservées par de nouvelles données ou utilisation de l'expiration de message.
    - Effacement administratif de la chaîne de rubrique.
  - Supprimez les données stockées sur un gestionnaire de files d'attente en supprimant l'intégralité du gestionnaire de files d'attente et les copies répliquées pour la haute disponibilité ou la reprise après incident.
  - Supprimez les données stockées par les commandes de trace de service en supprimant les fichiers dans le répertoire de trace.
  - Supprimez les données FFST stockées en supprimant les fichiers du répertoire des erreurs.
  - Supprimez les clichés d'espace adresse et d'unité de couplage (sous z/OS).
  - Supprimez l'archivage, la sauvegarde ou d'autres copies de ces données.
- Domaines de comportement IBM MQ à prendre en compte pour la conformité à la suppression des données de compte RGPD
  - Vous pouvez supprimer les données de compte et les préférences stockées par IBM MQ pour la connexion aux gestionnaires de files d'attente et aux services 3rd en les supprimant (y compris les copies d'archivage, de sauvegarde ou répliquées):
    - Objets d'informations d'authentification du gestionnaire de files d'attente qui stockent les données d'identification.

- Les enregistrements de droits d'accès du gestionnaire de files d'attente font référence à des identificateurs utilisateur.
- Règles d'authentification de canal de gestionnaire de files d'attente qui mappent ou bloquent des adresses IP spécifiques, des noms distinctifs de certificat ou des identificateurs d'utilisateur.
- Fichiers de données d'identification utilisés par l'agent IBM MQ Managed File Transfer , le consignateur et le plug-in MQ Explorer MFT pour l'authentification auprès du gestionnaire de files d'attente et des serveurs de fichiers.
- Certificats numériques X.509 qui représentent ou contiennent des informations sur un individu à partir de magasins de clés pouvant être utilisés par des connexions SSL/TLS ou IBM MQ Advanced Message Security (AMS).
- Comptes utilisateur individuels de IBM MQ Appliance, y compris la référence à ces comptes dans les fichiers journaux du système.
- Métadonnées de l'espace de travail IBM MQ Explorer et paramètres Eclipse .
- Fichier de stockage des mots de passe IBM MQ Explorer , comme indiqué dans les Préférences de mot de passe.
- Fichiers de configuration de la console IBM MQ et du serveur mqweb.
- Fichiers de configuration et magasins de clés IBM MQ Internet Pass-Thru .

Pour en savoir plus :

- [Authentification des connexions MFT et IBM MQ](#)
- [Mappage des données d'identification pour un serveur de fichiers à l'aide du fichier ProtocolBridgeCredentials.xml](#)
- [Configuration des utilisateurs et des rôles IBM MQ Console](#)

## Surveillance des données

IBM MQ fournit une série de fonctions de surveillance que les utilisateurs peuvent utiliser pour mieux comprendre les performances des applications et des gestionnaires de files d'attente.

IBM MQ fournit également un certain nombre de fonctions qui permettent de gérer les journaux d'erreurs du gestionnaire de files d'attente.

Pour en savoir plus :

- [Surveillance de votre réseau IBM MQ](#)
- [Services de messagerie de diagnostic](#)
- [ServiceQMErrorLog](#)
- [Surveillance et génération de rapports IBM MQ Appliance](#)

## Capacité de restreindre l'utilisation des données personnelles

A l'aide des fonctions récapitulées dans ce document, IBM MQ permet à un utilisateur final de restreindre l'utilisation de ses données personnelles.

Les files d'attente de messages IBM MQ ne doivent pas être utilisées comme magasin de données permanent de la même manière qu'une base de données, ce qui est particulièrement vrai lors du traitement des données d'application soumises au RGPD.

Contrairement à une base de données où des données peuvent être trouvées via une requête de recherche, il peut être difficile de trouver des données de message à moins que vous ne connaissiez les identificateurs de file d'attente, de message et de corrélation d'un message.

Les messages contenant les données d'un individu pouvant être facilement identifiés et localisés, il est possible d'utiliser les fonctions de messagerie IBM MQ standard pour accéder aux données de message ou les modifier.

## Gestion des fichiers

1. IBM MQ Managed File Transfer n'effectue pas d'analyse des logiciels malveillants sur les fichiers transférés. Les fichiers sont transférés tels qu'ils sont et un contrôle d'intégrité est effectué pour s'assurer que les données de fichier ne sont pas modifiées lors du transfert. Les totaux de contrôle source et cible sont publiés dans le cadre de la publication du statut de transfert. Il est recommandé que les utilisateurs finaux implémentent l'analyse des logiciels malveillants en fonction de leur environnement avant que MFT ne transfère le fichier et qu' MFT ne fournisse un fichier à un noeud final distant.
2. IBM MQ Managed File Transfer ne prend pas d'actions basées sur le type MIME ou l'extension de fichier. MFT lit les fichiers et transfère les octets exactement comme ils sont lus à partir du fichier d'entrée.

## Architectures basées sur un seul gestionnaire de files d'attente

---

Les architectures IBM MQ les plus simples impliquent la configuration et l'utilisation d'un seul gestionnaire de files d'attente.

Avant de planifier votre architecture IBM MQ , familiarisez-vous avec les concepts de base de IBM MQ . Voir [IBM MQ Technical overview](#).

Un certain nombre d'architectures possibles utilisant un seul gestionnaire de files d'attente sont décrites dans les sections suivantes:

- [«Gestionnaire de files d'attente unique avec des applications locales accédant à un service»](#), à la page [19](#)
- [«Gestionnaire de files d'attente unique avec des applications distantes accédant à un service en tant que clients»](#), à la page [19](#)
- [«Gestionnaire de files d'attente unique avec une configuration de publication / abonnement»](#), à la page [20](#)

### Gestionnaire de files d'attente unique avec des applications locales accédant à un service

La première architecture basée sur un gestionnaire de files d'attente unique est celle dans laquelle les applications accédant à un service s'exécutent sur le même système que les applications fournissant le service. Un gestionnaire de files d'attente IBM MQ fournit une intercommunication asynchrone entre les applications demandant le service et les applications fournissant le service. Cela signifie que la communication entre les applications peut se poursuivre même si l'une des applications est hors ligne pendant une période prolongée.

### Gestionnaire de files d'attente unique avec des applications distantes accédant à un service en tant que clients

Dans la deuxième architecture basée sur un seul gestionnaire de files d'attente, les applications s'exécutent à distance à partir des applications fournissant le service. Les applications distantes s'exécutent sur des systèmes différents des services. Les applications se connectent en tant que clients au gestionnaire de files d'attente unique. Cela signifie que l'accès à un service peut être fourni à plusieurs systèmes via un seul gestionnaire de files d'attente.

Une limitation de cette architecture est qu'une connexion réseau doit être disponible pour qu'une application puisse fonctionner. L'interaction entre l'application et le gestionnaire de files d'attente sur la connexion réseau est synchrone.

## Gestionnaire de files d'attente unique avec une configuration de publication / abonnement

Une autre architecture utilisant un gestionnaire de files d'attente unique consiste à utiliser une configuration de publication / abonnement. Dans la messagerie de publication / abonnement, vous pouvez découpler le fournisseur d'informations des consommateurs de ces informations. Cela diffère des styles de messagerie point à point dans les architectures précédemment décrites, où les applications doivent connaître les informations sur l'application cible, par exemple le nom de la file d'attente dans laquelle placer les messages. A l'aide de la publication / abonnement IBM MQ , l'application émettrice publie un message avec une rubrique spécifiée en fonction du sujet des informations. IBM MQ gère la distribution du message aux applications qui ont enregistré un intérêt dans ce sujet via un abonnement. Les applications de réception n'ont pas besoin de connaître la source des messages pour les recevoir. Pour plus d'informations, voir [Messagerie de publication / abonnement](#) et [Exemple de configuration de publication / abonnement de gestionnaire de files d'attente unique](#).

### Concepts associés

[Présentation de IBM MQ](#)

### Tâches associées

[«Planification d'une architecture IBM MQ», à la page 5](#)

Lors de la planification de l'environnement IBM MQ, prenez en compte la prise en charge fournie par IBM MQ pour les architectures à un ou plusieurs gestionnaires de files d'attente et pour les styles de messagerie de publication/abonnement et point à point. Prévoyez également vos exigences en matières de ressources et votre utilisation des fonctions de consignation et de sauvegarde.

[Création et gestion des gestionnaires de files d'attente sur Multiplatforms](#)

## Architectures basées sur plusieurs gestionnaires de files d'attente

Vous pouvez utiliser les techniques de mise en file d'attente de messages répartie pour créer une architecture IBM MQ impliquant la configuration et l'utilisation de plusieurs gestionnaires de files d'attente.

Avant de planifier votre architecture IBM MQ , familiarisez-vous avec les concepts de base de IBM MQ . Voir [IBM MQ Technical overview](#).

Une architecture IBM MQ peut être modifiée, sans modification des applications qui fournissent des services, en ajoutant des gestionnaires de files d'attente supplémentaires.

Les applications peuvent être hébergées sur la même machine qu'un gestionnaire de files d'attente, puis obtenir une communication asynchrone avec un service hébergé sur un autre gestionnaire de files d'attente sur un autre système. Les applications accédant à un service peuvent également se connecter en tant que clients à un gestionnaire de files d'attente qui fournit ensuite un accès asynchrone au service sur un autre gestionnaire de files d'attente.

Les routes qui connectent différents gestionnaires de files d'attente et leurs files d'attente sont définies à l'aide de techniques de mise en file d'attente répartie. Les gestionnaires de files d'attente de l'architecture sont connectés à l'aide de canaux. Les canaux sont utilisés pour déplacer automatiquement des messages d'un gestionnaire de files d'attente vers un autre dans une direction en fonction de la configuration des gestionnaires de files d'attente.

Pour une présentation générale de la planification d'un réseau IBM MQ , voir [«Conception de réseaux de gestionnaires de files d'attente réparties», à la page 22](#).

Pour plus d'informations sur la planification des canaux pour votre architecture IBM MQ , voir [IBM MQ distributed queuing techniques](#).

La gestion de files d'attente réparties vous permet de créer et de surveiller les communications entre les gestionnaires de files d'attente. Pour plus d'informations sur la gestion répartie des files d'attente, voir [Présentation de la gestion répartie des files d'attente](#).

### Tâches associées

[«Planification d'une architecture IBM MQ», à la page 5](#)

Lors de la planification de l'environnement IBM MQ, prenez en compte la prise en charge fournie par IBM MQ pour les architectures à un ou plusieurs gestionnaires de files d'attente et pour les styles de messagerie de publication/abonnement et point à point. Prévoyez également vos exigences en matières de ressources et votre utilisation des fonctions de consignation et de sauvegarde.

[Création et gestion des gestionnaires de files d'attente sur Multiplatforms](#)

## Planification de vos files d'attente réparties et de vos clusters

Vous pouvez connecter manuellement des files d'attente hébergées sur des gestionnaires de files d'attente réparties ou créer un cluster de gestionnaires de files d'attente et laisser le produit connecter les gestionnaires de files d'attente pour vous. Pour choisir une topologie adaptée à votre réseau de messagerie distribué, vous devez prendre en compte vos exigences en matière de contrôle manuel, de taille du réseau, de fréquence de changement, de disponibilité et d'évolutivité.

### Avant de commencer

Cette tâche suppose que vous comprenez ce qu'est un réseau de messagerie distribué et comment il fonctionne. Pour une présentation technique, voir [Mise en file d'attente répartie et clusters](#).

### Pourquoi et quand exécuter cette tâche

Pour créer un réseau de messagerie réparti, vous pouvez configurer manuellement des canaux pour connecter des files d'attente hébergées sur différents gestionnaires de files d'attente ou créer un cluster de gestionnaires de files d'attente. La mise en cluster permet aux gestionnaires de files d'attente de communiquer entre eux sans qu'il soit nécessaire de configurer des définitions de canal supplémentaires ou des définitions de file d'attente éloignée, ce qui simplifie leur configuration et leur gestion.

Pour choisir une topologie adaptée à votre réseau de publication / abonnement distribué, vous devez prendre en compte les questions générales suivantes:

- De combien de contrôle manuel avez-vous besoin sur les connexions de votre réseau?
- Quelle sera la taille de votre réseau?
- Quelle sera sa dynamique?
- Quelles sont vos exigences en matière de disponibilité et d'évolutivité?

### Procédure

- Prenez en compte le contrôle manuel dont vous avez besoin sur les connexions de votre réseau.

Si vous n'avez besoin que de quelques connexions ou si des connexions individuelles doivent être définies de manière très précise, vous devez probablement créer le réseau manuellement.

Si vous avez besoin de plusieurs gestionnaires de files d'attente qui sont liés de manière logique et qui doivent partager des données et des applications, vous devez envisager de les regrouper dans un cluster de gestionnaires de files d'attente.

- Estimez la taille de votre réseau.
  - a) Estimez le nombre de gestionnaires de files d'attente dont vous avez besoin. N'oubliez pas que les files d'attente peuvent être hébergées sur plusieurs gestionnaires de files d'attente.
  - b) Si vous envisagez d'utiliser un cluster, ajoutez deux gestionnaires de files d'attente supplémentaires pour agir en tant que référentiels complets.

Pour les réseaux de plus grande taille, la configuration manuelle et la maintenance des connexions peuvent prendre beaucoup de temps, et vous devez envisager d'utiliser un cluster.

- Tenez compte de la dynamique de l'activité du réseau.

Planifiez l'hébergement des files d'attente occupées sur les gestionnaires de files d'attente performants.

Si vous prévoyez de créer et de supprimer fréquemment des files d'attente, envisagez d'utiliser un cluster.

- Tenez compte de vos exigences en matière de disponibilité et d'évolutivité.
  - a) Décidez si vous devez garantir la haute disponibilité des gestionnaires de files d'attente. Si tel est le cas, estimez le nombre de gestionnaires de files d'attente auxquels cette exigence s'applique.
  - b) Déterminez si certains de vos gestionnaires de files d'attente sont moins capables que d'autres.
  - c) Déterminez si les liens de communication vers certains de vos gestionnaires de files d'attente sont plus fragiles que vers d'autres.
  - d) Envisagez d'héberger des files d'attente sur plusieurs gestionnaires de files d'attente.

Les réseaux et les clusters configurés manuellement peuvent être configurés pour être hautement disponibles et évolutifs. Si vous utilisez un cluster, vous devez définir deux gestionnaires de files d'attente supplémentaires en tant que référentiels complets. Le fait de disposer de deux référentiels complets garantit que le cluster continue de fonctionner si l'un des référentiels complets devient indisponible. Assurez-vous que les gestionnaires de files d'attente de référentiel complet sont robustes, performants et disposent d'une bonne connectivité réseau. N'envisagez pas d'utiliser les gestionnaires de files d'attente de référentiel complet pour d'autres travaux.

- En fonction de ces calculs, utilisez les liens fournis pour vous aider à choisir entre la configuration manuelle des connexions entre les gestionnaires de files d'attente ou l'utilisation d'un cluster.

## Que faire ensuite

Vous êtes maintenant prêt à configurer votre réseau de messagerie distribué.

### Tâches associées

[Configuration de la file d'attente répartie](#)

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

## Conception de réseaux de gestionnaires de files d'attente réparties

IBM MQ envoie et reçoit des données entre des applications et sur des réseaux à l'aide de gestionnaires de files d'attente et de canaux. La planification de réseau implique la définition des exigences pour créer une infrastructure permettant de connecter ces systèmes sur un réseau.

Des canaux peuvent être créés entre votre système et tout autre système avec lequel vous devez communiquer. Des canaux multitransons peuvent être créés pour se connecter à des systèmes sur lesquels vous n'avez aucune connexion directe. Les connexions de canal de transmission de messages décrites dans les scénarios sont présentées sous la forme d'un diagramme réseau dans [Figure 1](#), à la [page 23](#).

Si vous devez créer des canaux entre des systèmes sur des réseaux physiques différents ou des canaux qui communiquent via un pare-feu, l'utilisation de IBM MQ Internet Pass-Thru peut simplifier la configuration. Pour plus d'informations, voir [IBM MQ Internet Pass-Thru](#).

## Noms de file d'attente de transmission et de canal

Les files d'attente de transmission peuvent recevoir n'importe quel nom. Toutefois, pour éviter toute confusion, vous pouvez leur attribuer les mêmes noms que les noms de gestionnaire de files d'attente de destination ou les noms d'alias de gestionnaire de files d'attente, selon le cas. La file d'attente de transmission est ainsi associée à la route qu'elle utilise, ce qui donne une vue d'ensemble claire des routes parallèles créées via des gestionnaires de files d'attente intermédiaires (multitransons).

Il n'est pas si clair pour les noms de canal. Les noms de canal dans [Figure 1](#), à la [page 23](#) pour QM2, par exemple, doivent être différents pour les canaux entrants et sortants. Tous les noms de canal peuvent toujours contenir leurs noms de file d'attente de transmission, mais ils doivent être qualifiés pour les rendre uniques.

Par exemple, dans QM2, il existe un canal QM3 provenant de QM1 et un canal QM3 allant à QM3. Pour rendre les noms uniques, le premier peut être nommé QM3\_from\_QM1 et le second peut être nommé QM3\_from\_QM2. De cette manière, les noms de canal affichent le nom de la file d'attente de transmission

dans la première partie du nom. La direction et le nom du gestionnaire de files d'attente adjacent sont affichés dans la deuxième partie du nom.

Un tableau des noms de canal suggérés pour [Figure 1](#), à la [page 23](#) est fourni dans [Tableau 1](#), à la [page 23](#).

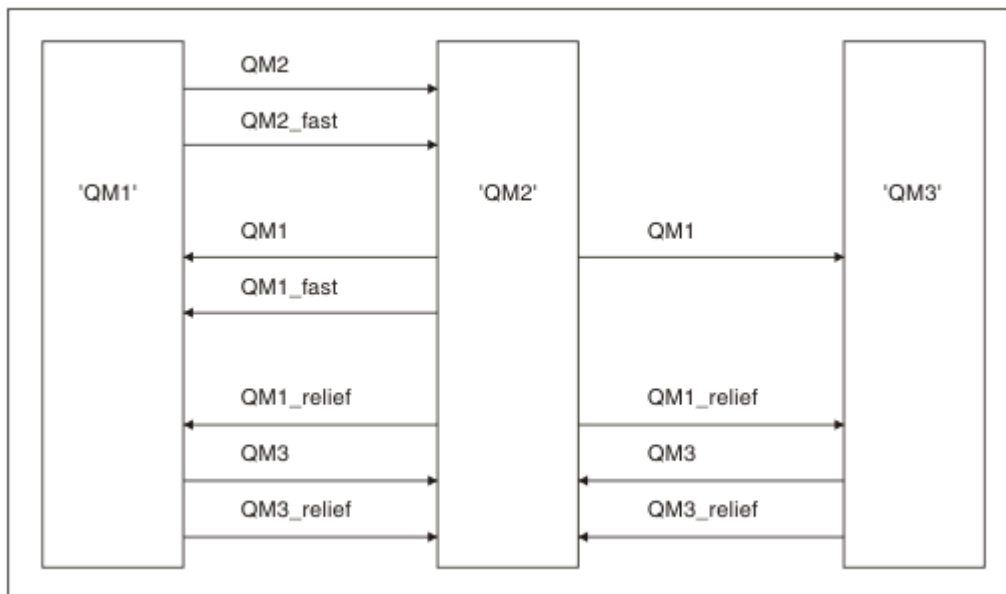



Figure 1. Diagramme réseau montrant tous les canaux

Tableau 1. Exemple de noms de canal

Nom du chemin	Gestionnaires de files d'attente hébergeant le canal	Nom de la file d'attente de transmission	Nom de canal suggéré
QM1	QM1 & QM2	QM1 (à l'adresse QM2)	QM1.from.QM2
QM1	QM2 & QM3	QM1 (à l'adresse QM3)	QM1.from.QM3
QM1_fast	QM1 & QM2	QM1_fast (à l'adresse QM2)	QM1_fast.from.QM2
QM1_relief	QM1 & QM2	QM1_relief (à l'adresse QM2)	QM1_relief.from.QM2
QM1_relief	QM2 & QM3	QM1_relief (à l'adresse QM3)	QM1_relief.from.QM3
QM2	QM1 & QM2	QM2 (à l'adresse QM1)	QM2.from.QM1
QM2_fast	QM1 & QM2	QM2_fast (à l'adresse QM1)	QM2_fast.from.QM1
QM3	QM1 & QM2	QM3 (sur QM1)	QM3.from.QM1
QM3	QM2 & QM3	QM3 (à l'adresse QM2)	QM3.from.QM2
QM3_relief	QM1 & QM2	QM3_relief (à l'adresse QM1)	QM3_relief.from.QM1
QM3_relief	QM2 & QM3	QM3_relief (à l'adresse QM2)	QM3_relief.from.QM2

**Remarque :**

1.  Sous IBM MQ for z/OS, les noms de gestionnaire de files d'attente sont limités à quatre caractères.

2. Nommez tous les canaux de votre réseau de manière unique. Comme indiqué dans la [Tableau 1](#), à la [page 23](#), l'inclusion des noms de gestionnaire de files d'attente source et cible dans le nom de canal est un bon moyen de le faire.

## Planificateur de réseau

La création d'un réseau suppose qu'il existe une autre fonction de niveau supérieur du *planificateur de réseau* dont les plans sont implémentés par les autres membres de l'équipe.

Pour les applications largement utilisées, il est plus économique de penser en termes de sites d'accès locaux pour la concentration du trafic de messages, en utilisant des liens à large bande entre les sites d'accès locaux, comme illustré dans la [Figure 2](#), à la [page 25](#).

Dans cet exemple, il y a deux systèmes principaux et un certain nombre de systèmes satellites. La configuration réelle dépend des considérations métier. Deux gestionnaires de files d'attente de concentrateur se trouvent dans des centres pratiques. Chaque concentrateur QM possède des canaux de message vers les gestionnaires de files d'attente locaux:

- Le concentrateur QM 1 dispose de canaux de transmission de messages vers chacun des trois gestionnaires de files d'attente locaux, QM1, QM2 et QM3. Les applications qui utilisent ces gestionnaires de files d'attente peuvent communiquer entre elles via les concentrateurs QM.
- Le concentrateur 2 de gestionnaire de files d'attente comporte des canaux de transmission de messages vers chacun des trois gestionnaires de files d'attente locaux, QM4, QM5 et QM6. Les applications qui utilisent ces gestionnaires de files d'attente peuvent communiquer entre elles via les concentrateurs QM.
- Les concentrateurs QM disposent de canaux de messages entre eux, ce qui permet à toute application d'un gestionnaire de files d'attente d'échanger des messages avec toute autre application d'un autre gestionnaire de files d'attente.



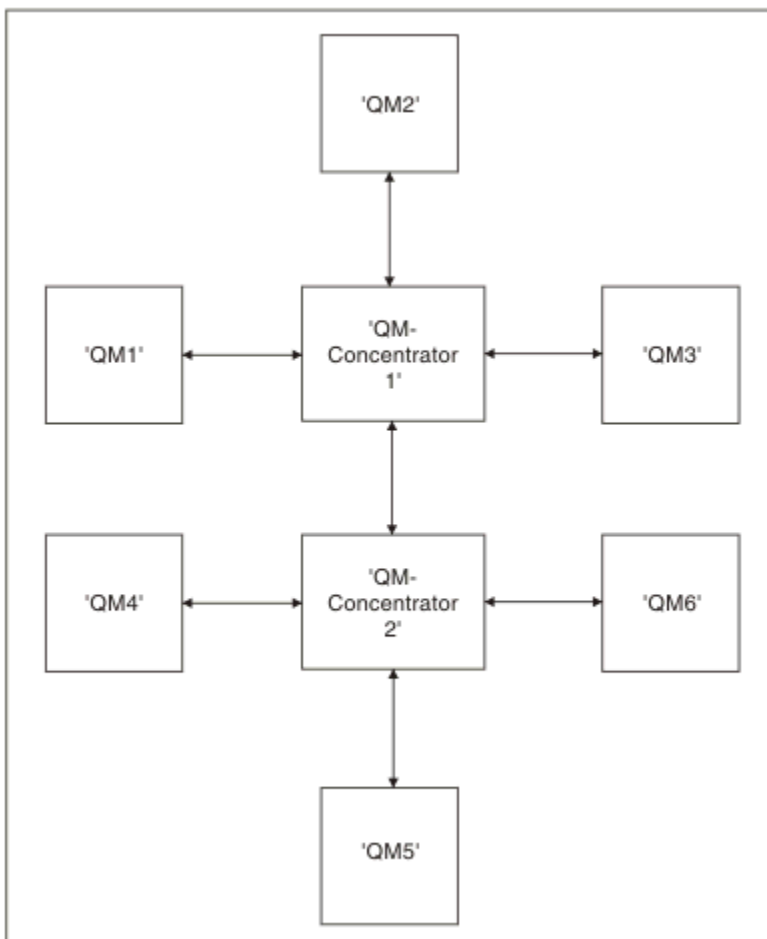


Figure 2. Diagramme réseau montrant des concentrateurs QM

## Conception des clusters

Les clusters fournissent un mécanisme d'interconnexion des gestionnaires de files d'attente qui simplifie à la fois la configuration initiale et la gestion en cours. Les clusters doivent être soigneusement conçus pour s'assurer qu'ils fonctionnent correctement et qu'ils atteignent les niveaux de disponibilité et de réactivité requis.

### Avant de commencer


Pour une introduction aux concepts de mise en cluster, voir les rubriques suivantes:

- [Mise en file d'attente répartie et clusters](#)
- [«Comparaison de la mise en cluster et de la mise en file d'attente répartie», à la page 32](#)
- [Composants d'un cluster](#)

Lorsque vous concevez votre cluster de gestionnaires de files d'attente, vous devez prendre des décisions. Vous devez d'abord décider quels gestionnaires de files d'attente du cluster doivent contenir les référentiels complets des informations de cluster. Tout gestionnaire de files d'attente que vous créez peut fonctionner dans un cluster. Vous pouvez choisir n'importe quel nombre de gestionnaires de files d'attente à cette fin, mais le nombre idéal est deux. Pour plus d'informations sur la sélection des gestionnaires de files d'attente devant contenir les référentiels complets, voir [«Comment choisir des gestionnaires de files d'attente de cluster pour stocker des référentiels complets», à la page 34.](#)

Pour plus d'informations sur la conception de votre cluster, voir les rubriques suivantes:

- [«Exemples de clusters», à la page 41](#)

- «Organisation d'un cluster», à la page 36
- «Conventions de dénomination de cluster», à la page 36
-  «Queue sharing groups and clusters», à la page 38
- «Chevauchement de clusters», à la page 38


## Que faire ensuite

Pour plus d'informations sur la configuration et l'utilisation des clusters, voir les rubriques suivantes:

- [Etablissement d'une communication dans un cluster](#)
- [Configuration d'un cluster de gestionnaires de files d'attente](#)
- [Acheminement des messages vers et depuis les clusters](#)
- [Utilisation des clusters pour la gestion de la charge de travail](#)

Pour plus d'informations sur la configuration de votre cluster, voir [«Conseils de mise en cluster»](#), à la page 39.

## Planification de l'utilisation de plusieurs files d'attente de transmission de cluster

Vous pouvez définir explicitement des files d'attente de transmission ou demander au système de les générer pour vous. Si vous définissez les files d'attente de transmission vous-même, vous avez plus de contrôle sur les définitions de file d'attente.  Sous z/OS, vous avez également plus de contrôle sur l'ensemble de pages dans lequel les messages sont conservés.

## Définition des files d'attente de transmission


Il existe deux méthodes pour définir des files d'attente de transmission:

- Automatiquement, à l'aide de l'attribut de gestionnaire de files d'attente DEFCLXQ, comme suit:

```
ALTER QMGR DEFCLXQ(SCTQ | CHANNEL)
```

DEFCLXQ (SCTQ) indique que la file d'attente de transmission par défaut pour tous les canaux émetteurs de cluster est SYSTEM.CLUSTER.TRANSMIT.QUEUE. Il s'agit de la valeur par défaut.

DEFCLXQ (CHANNEL) indique que par défaut, chaque canal émetteur de cluster utilise une file d'attente de transmission distincte nommée SYSTEM.CLUSTER.TRANSMIT.*nom de canal*. Chaque file d'attente de transmission est automatiquement définie par le gestionnaire de files d'attente. Pour plus d'informations, voir [«Files d'attente de transmission de cluster définies automatiquement»](#), à la page 28.

- Manuellement, en définissant une file d'attente de transmission avec une valeur spécifiée pour l'attribut CLCHNAME. L'attribut CLCHNAME indique les canaux émetteurs de cluster qui doivent utiliser la file d'attente de transmission.  Si vous définissez manuellement une file d'attente de transmission sur z/OS, voir [«Planification des files d'attente de transmission de cluster définies manuellement»](#), à la page 29 pour plus d'informations.

## De quelle sécurité ai-je besoin?

Pour lancer un commutateur, automatiquement ou manuellement, vous devez disposer des droits permettant de démarrer un canal.

Pour définir la file d'attente utilisée en tant que file d'attente de transmission, vous devez disposer des droits IBM MQ standard pour définir la file d'attente.

## Quand le moment est-il opportun pour mettre en oeuvre le changement?

Lors de la modification de la file d'attente de transmission utilisée par les canaux émetteurs de cluster, vous devez allouer un délai pour effectuer la mise à jour, en tenant compte des points suivants:

- Le temps nécessaire à un canal pour changer de file d'attente de transmission dépend du nombre total de messages dans l'ancienne file d'attente de transmission, du nombre de messages à déplacer et de la taille des messages.
- Les applications peuvent continuer à insérer des messages dans la file d'attente de transmission pendant la modification. Cela peut entraîner une augmentation du temps de transition.
- Vous pouvez modifier le paramètre CLCHNAME de n'importe quelle file d'attente de transmission ou DEFCLXQ à n'importe quel moment, de préférence lorsque la charge de travail est faible.

Notez que rien ne se passe immédiatement.

- Les modifications se produisent uniquement lorsqu'un canal démarre ou redémarre. Lorsqu'un canal démarre, il vérifie la configuration en cours et passe à une nouvelle file d'attente de transmission si nécessaire.
- Plusieurs modifications peuvent modifier l'association d'un canal émetteur de cluster à une file d'attente de transmission:
  - Modification de la valeur de l'attribut CLCHNAME d'une file d'attente de transmission, rendant CLCHNAME moins spécifique ou à blanc.
  - Modification de la valeur de l'attribut CLCHNAME d'une file d'attente de transmission, ce qui rend CLCHNAME plus spécifique.
  - Suppression d'une file d'attente avec CLCHNAME indiqué.
  - Modification de l'attribut de gestionnaire de files d'attente DEFCLXQ.


## Combien de temps l'interrupteur prendra-t-il?

Pendant la période de transition, tous les messages du canal sont déplacés d'une file d'attente de transmission à une autre. Le temps nécessaire à un canal pour changer de file d'attente de transmission dépend du nombre total de messages dans l'ancienne file d'attente de transmission et du nombre de messages à déplacer.

Pour les files d'attente contenant quelques milliers de messages, le déplacement des messages doit prendre moins d'une seconde. La durée réelle dépend du nombre et de la taille des messages. Votre gestionnaire de files d'attente doit pouvoir déplacer des messages de plusieurs mégaoctets par seconde.

Les applications peuvent continuer à insérer des messages dans la file d'attente de transmission pendant la modification. Cela peut entraîner une augmentation du temps de transition.

Chaque canal émetteur de cluster affecté doit être redémarré pour que la modification soit prise en compte. Par conséquent, il est préférable de modifier la configuration de la file d'attente de transmission lorsque le gestionnaire de files d'attente n'est pas occupé et que peu de messages sont stockés dans les files d'attente de transmission du cluster.

La **runswchl** commande  ou la commande SWITCH CHANNEL (\*) STATUS dans CSQTIL sur z/OS peut être utilisée pour interroger le statut des canaux émetteurs de cluster et les modifications en attente de leur configuration de file d'attente de transmission.

## Comment implémenter le changement

Voir [Implémentation du système à l'aide de plusieurs files d'attente de transmission de cluster](#) pour plus de détails sur la manière d'apporter des modifications à plusieurs files d'attente de transmission de cluster, automatiquement ou manuellement.

## Annulation de la modification




Voir [Annulation d'une modification apportée à une file d'attente de transmission sur z/OS](#) pour plus de détails sur l'annulation des modifications si vous rencontrez des problèmes.

*Files d'attente de transmission de cluster définies automatiquement*


Vous pouvez demander au système de générer automatiquement les files d'attente de transmission.

## Avant de commencer

 Pour configurer manuellement les files d'attente de transmission du cluster sur z/OS, voir «[Planification des files d'attente de transmission de cluster définies manuellement](#)», à la page 29.

## Pourquoi et quand exécuter cette tâche

Si aucun canal n'est associé à une file d'attente de transmission de cluster définie manuellement et que vous spécifiez DEFCLXQ (CHANNEL), lorsque le canal démarre, le gestionnaire de files d'attente définit automatiquement une file d'attente dynamique permanente pour le canal émetteur de cluster. File d'attente modèle SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE est utilisé pour définir automatiquement la file d'attente de transmission de cluster dynamique permanente avec le nom SYSTEM.CLUSTER.TRANSMIT.Channe1Name.

**Important :**  Dans IBM MQ 8.0, le gestionnaire de files d'attente n'a pas le SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE. Vous ne pouvez pas migrer directement depuis IBM MQ 8.0 à cette version. Pour plus d'informations sur l'ajout du SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE dans un gestionnaire de files d'attente migré depuis IBM MQ 8.0, consultez cette rubrique dans la documentation de la version intermédiaire que vous avez utilisée pour migrer le gestionnaire de files d'attente.

## Procédure

1. Utilisez l'attribut de gestionnaire de files d'attente *DEFCLXQ*.

Pour plus d'informations sur cet attribut, voir [ALTER QMGR](#).

Deux options sont possibles :

### Sctq

Cette option est la valeur par défaut et signifie que vous utilisez le système SYSTEM.CLUSTER.TRANSMIT.QUEUE.

### Canal

Signifie que vous utilisez plusieurs files d'attente de transmission de cluster.

2. Pour passer à la nouvelle association:

- Arrêtez et redémarrez le canal.
- Le canal utilise la nouvelle définition de file d'attente de transmission.
- Les messages sont transférés par un processus de basculement transitoire de l'ancienne file d'attente vers la nouvelle file d'attente de transmission.

Notez que tous les messages d'application sont insérés dans l'ancienne définition.

Lorsque le nombre de messages de l'ancienne file d'attente atteint zéro, les nouveaux messages sont placés directement dans la nouvelle file d'attente de transmission.

3. Pour surveiller la fin du processus de commutation:

- a) Un commutateur de file d'attente de transmission lancé par un canal s'exécute en arrière-plan et votre administrateur peut surveiller le journal des travaux du gestionnaire de files d'attente pour déterminer quand il est terminé.
- b) Surveillez les messages de l'historique du travail pour afficher la progression du commutateur.

c) Pour vous assurer que seuls les canaux de votre choix utilisent cette file d'attente de transmission, émettez la commande DIS CLUSQMGR (\*) où, par exemple, la propriété de file d'attente de transmission qui définit la file d'attente de transmission est APPQMGR . CLUSTER1 . XMITQ.

d) 

Utilisez la commande SWITCH CHANNEL (\*) STATUS sous CSQUTIL.

Cette option vous indique les modifications en attente et le nombre de messages à déplacer entre les files d'attente de transmission.

## Résultats

Vous avez configuré votre ou vos files d'attente de transmission de cluster.

### Tâches associées

«Planification des files d'attente de transmission de cluster définies manuellement», à la page 29  
Sous IBM MQ for z/OS, si vous définissez vous-même les files d'attente de transmission, vous avez davantage de contrôle sur les définitions et sur l'ensemble de pages sur lequel les messages sont conservés.

### Référence associée

[ALTER QMGR](#)

[DISPLAY CLUSQMGR](#)



*Planification des files d'attente de transmission de cluster définies manuellement*

Sous IBM MQ for z/OS, si vous définissez vous-même les files d'attente de transmission, vous avez davantage de contrôle sur les définitions et sur l'ensemble de pages sur lequel les messages sont conservés.

## Avant de commencer

Pour configurer automatiquement les files d'attente de transmission de cluster, voir [«Files d'attente de transmission de cluster définies automatiquement»](#), à la page 28.

## Pourquoi et quand exécuter cette tâche

Votre administrateur définit manuellement une file d'attente de transmission et utilise l'attribut de file d'attente CLCHNAME pour définir le ou les canaux émetteurs de cluster qui utiliseront cette file d'attente comme file d'attente de transmission.

Notez que CLCHNAME peut inclure un caractère générique au début ou à la fin pour permettre l'utilisation d'une seule file d'attente pour plusieurs canaux.

## Procédure

1. Par exemple, entrez ce qui suit :

```
DEFINE QLOCAL (APPQMGR . CLUSTER1 . XMITQ)
CLCHNAME (CLUSTER1 . TO . APPQMGR)
USAGE (XMITQ) STGCLASS (STG1)
INDXTYPE ( CORRELID ) SHARE

DEFINE STGCLASS (STG1) PSID (3)
DEFINE PSID (3) BUFFERPOOL (4)
```

**Conseil :** Vous devez planifier l'ensemble de pages (et le pool de mémoire tampon) que vous utilisez pour vos files d'attente de transmission. Vous pouvez avoir des ensembles de pages différents pour différentes files d'attente et les isoler les uns des autres. Par conséquent, le remplissage d'un ensemble de pages n'a pas d'impact sur les files d'attente de transmission des autres ensembles de pages.

Pour plus d'informations sur la façon dont chaque canal sélectionne la file d'attente appropriée, voir [Utilisation des files d'attente de transmission de cluster et des canaux émetteurs de cluster](#) .

Lorsque le canal démarre, il bascule son association vers la nouvelle file d'attente de transmission. Pour s'assurer qu'aucun message n'est perdu, le gestionnaire de files d'attente transfère automatiquement les messages de l'ancienne file d'attente de transmission du cluster vers la nouvelle file d'attente de transmission dans l'ordre.

2. Utilisez la fonction CSQUTIL SWITCH pour passer à la nouvelle association.

Pour plus d'informations, voir [Commutation de la file d'attente de transmission associée aux canaux émetteurs de cluster \(SWITCH\)](#) .

- a) Arrêtez le ou les canaux dont la file d'attente de transmission doit être modifiée pour qu'ils soient à l'état ARRETE.

Exemple :

```
STOP CHANNEL (CLUSTER1 .TO .APPQMGR)
```

- b) Modifiez l'attribut CLCHNAME (XXXX) dans la file d'attente de transmission.
- c) Utilisez la fonction SWITCH pour changer les messages ou surveiller ce qui se passe.

Utilisez la commande

```
SWITCH CHANNEL (*) MOVEMSGS (YES)
```

pour déplacer les messages sans démarrer le canal.

- d) Démarrez le ou les canaux et vérifiez si le canal utilise les files d'attente correctes.

Exemple :

```
DIS CHS (CLUSTER1 .TO .APPQMGR)
DIS CHS (*) where (XMITQ eq APPQMGR .CLUSTER1 .XMITQ)
```

**Conseil :** Le processus suivant utilise la fonction CSQUTIL SWITCH. Pour plus d'informations, voir [Permutation de la file d'attente de transmission associée aux canaux émetteurs de cluster \(SWITCH\)](#).

Vous n'avez pas besoin d'utiliser cette fonction, mais l'utilisation de cette fonction offre plus d'options:

- L'utilisation de SWITCH CHANNEL (\*) STATUS permet d'identifier facilement le statut de commutation des canaux émetteurs de cluster. Il permet à votre administrateur de voir quels canaux commutent actuellement, et les canaux dont le commutateur est en attente qui prennent effet lors du prochain démarrage de ces canaux.

Sans cette fonction, votre administrateur doit utiliser plusieurs commandes DISPLAY, puis traiter la sortie résultante pour vérifier ces informations. Votre administrateur peut également confirmer qu'un changement de configuration a le résultat requis.

- Si CSQUTIL est utilisé pour lancer le commutateur, CSQUTIL continue de surveiller la progression de cette opération et ne s'arrête que lorsque le commutateur est terminé.

Cela peut faciliter l'exécution de ces opérations par lots. De plus, si CSQUTIL est exécuté pour commuter plusieurs canaux, CSQUTIL effectue ces actions de manière séquentielle ; cela peut avoir moins d'impact pour votre entreprise que plusieurs commutateurs s'exécutant en parallèle.

## Résultats

Vous avez configuré votre ou vos files d'attente de transmission de cluster sur z/OS.

### **Contrôle d'accès et files d'attente de transmission de cluster multiples**

Choisissez entre trois modes de vérification lorsqu'une application place des messages dans des files d'attente de cluster éloignées. Les modes sont la vérification à distance par rapport à la file d'attente de

cluster, la vérification en local par rapport à `SYSTEM.CLUSTER.TRANSMIT.QUEUE` ou la vérification par rapport aux profils locaux pour la file d'attente de cluster ou le gestionnaire de files d'attente de cluster.


IBM MQ vous permet de vérifier localement, ou localement et à distance, qu'un utilisateur est autorisé à placer un message dans une file d'attente éloignée. Une application IBM MQ standard utilise uniquement la vérification locale et s'appuie sur le gestionnaire de files d'attente éloignées pour approuver les vérifications d'accès effectuées sur le gestionnaire de files d'attente local. Si la vérification à distance n'est pas utilisée, le message est placé dans la file d'attente cible avec les droits du processus de canal de transmission de messages éloigné. Pour utiliser la vérification à distance, vous devez définir les droits d'insertion du canal récepteur sur la sécurité du contexte.

Les vérifications locales sont effectuées sur la file d'attente ouverte par l'application. Dans la mise en file d'attente répartie, l'application ouvre généralement une définition de file d'attente éloignée et des contrôles d'accès sont effectués sur la définition de file d'attente éloignée. Si le message est inséré avec un en-tête de routage complet, les vérifications sont effectuées sur la file d'attente de transmission. Si une application ouvre une file d'attente de cluster qui ne se trouve pas sur le gestionnaire de files d'attente local, il n'y a pas d'objet local à vérifier. Les vérifications de contrôle d'accès sont effectuées sur la file d'attente de transmission du cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Même avec plusieurs files d'attente de transmission de cluster, les vérifications de contrôle d'accès local pour les files d'attente de cluster éloignées sont effectuées sur `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Le choix de la vérification locale ou à distance est un choix entre deux extrêmes. La vérification à distance est à granularité fine. Chaque utilisateur doit disposer d'un profil de contrôle d'accès sur chaque gestionnaire de files d'attente du cluster pour pouvoir le placer dans une file d'attente de cluster. La vérification locale est à granularité grossière. Chaque utilisateur n'a besoin que d'un seul profil de contrôle d'accès pour la file d'attente de transmission de cluster sur le gestionnaire de files d'attente auquel il est connecté. Avec ce profil, ils peuvent placer un message dans n'importe quelle file d'attente de cluster sur n'importe quel gestionnaire de files d'attente de n'importe quel cluster.

Les administrateurs disposent d'un autre moyen de configurer le contrôle d'accès pour les files d'attente de cluster. Vous pouvez créer un profil de sécurité pour une file d'attente de cluster sur n'importe quel gestionnaire de files d'attente du cluster à l'aide de la commande `setmqaut`. Le profil prend effet si vous ouvrez une file d'attente de cluster éloignée en local, en spécifiant uniquement le nom de la file d'attente. Vous pouvez également configurer un profil pour un gestionnaire de files d'attente éloignées. Dans ce cas, le gestionnaire de files d'attente peut vérifier le profil d'un utilisateur qui ouvre une file d'attente de cluster en fournissant un nom qualifié complet.

Les nouveaux profils ne fonctionnent que si vous modifiez la strophe du gestionnaire de files d'attente, **ClusterQueueAccessControl** en RQMName. La valeur par défaut est `Xmitq`. Vous devez créer des profils pour toutes les files d'attente de cluster que les applications existantes utilisent. Si vous remplacez la strophe par RQMName sans créer de profils, les applications risquent d'échouer.

**Conseil :** La vérification de l'accès à la file d'attente de cluster ne s'applique pas à la mise en file d'attente éloignée. Des contrôles d'accès sont toujours effectués sur les définitions locales. Les modifications signifient que vous pouvez suivre la même approche pour configurer la vérification d'accès sur les files d'attente de cluster et les rubriques de cluster.  Les modifications permettent également d'aligner plus étroitement l'approche de contrôle d'accès pour les files d'attente de cluster avec z/OS. Les commandes de configuration de la vérification d'accès sur z/OS sont différentes, mais les deux vérifient l'accès sur un profil plutôt que sur l'objet lui-même.

### Concepts associés

[«Mise en cluster: isolement d'application à l'aide de plusieurs files d'attente de transmission de cluster», à la page 51](#)

Vous pouvez isoler les flux de messages entre les gestionnaires de files d'attente d'un cluster. Vous pouvez placer des messages transportés par des canaux émetteurs de cluster différents dans des files d'attente de transmission de cluster différentes. Vous pouvez utiliser l'approche dans un cluster unique ou avec des clusters qui se chevauchent. La rubrique fournit des exemples et des meilleures pratiques pour vous guider dans le choix d'une approche à utiliser.

### Tâches associées

[Paramètre ClusterQueueAccessControl](#)

## Comparaison de la mise en cluster et de la mise en file d'attente répartie

Comparez les composants qui doivent être définis pour connecter les gestionnaires de files d'attente à l'aide de la mise en file d'attente répartie et de la mise en cluster.

Si vous n'utilisez pas de clusters, vos gestionnaires de files d'attente sont indépendants et communiquent à l'aide de la mise en file d'attente répartie. Si un gestionnaire de files d'attente doit envoyer des messages à un autre, vous devez définir:

- Une file d'attente de transmission
- Un canal vers le gestionnaire de files d'attente éloignées

La Figure 3, à la page 32 présente les composants requis pour la mise en file d'attente répartie.

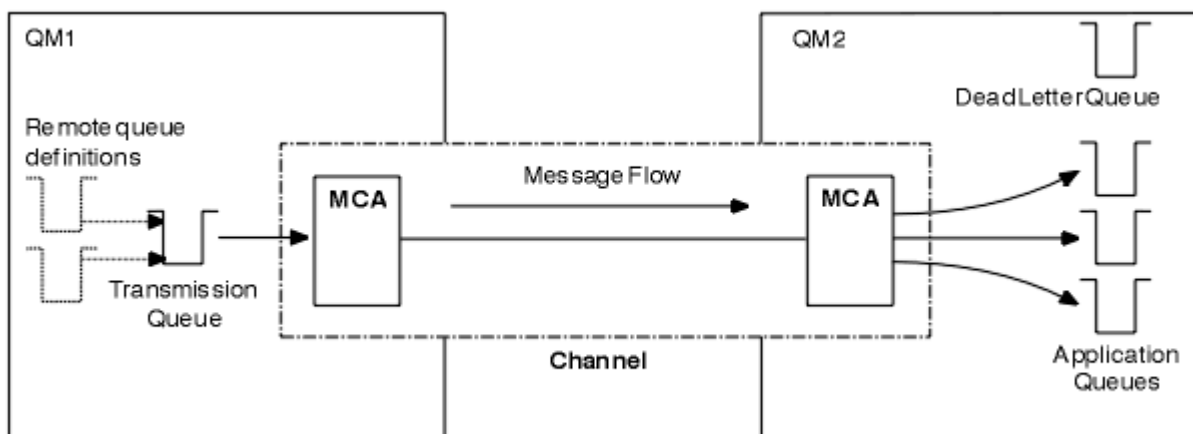


Figure 3. Files d'attente réparties

Si vous regroupez des gestionnaires de files d'attente dans un cluster, les files d'attente de n'importe quel gestionnaire de files d'attente sont disponibles pour n'importe quel autre gestionnaire de files d'attente du cluster. Tout gestionnaire de files d'attente peut envoyer un message à n'importe quel autre gestionnaire de files d'attente du même cluster sans définitions explicites. Vous ne fournissez pas de définitions de canal, de définitions de file d'attente éloignée ou de files d'attente de transmission pour chaque destination. Chaque gestionnaire de files d'attente d'un cluster possède une file d'attente de transmission unique à partir de laquelle il peut transmettre des messages à n'importe quel autre gestionnaire de files d'attente du cluster. Chaque gestionnaire de files d'attente d'un cluster doit uniquement définir:

- Un canal récepteur de cluster sur lequel recevoir les messages
- Un canal émetteur de cluster avec lequel il se présente et apprend à connaître le cluster

## Définitions permettant de configurer un cluster par rapport à la mise en file d'attente répartie

Consultez la rubrique Figure 4, à la page 33, qui présente quatre gestionnaires de files d'attente comportant chacun deux files d'attente. Tenez compte du nombre de définitions nécessaires pour connecter ces gestionnaires de files d'attente à l'aide de la mise en file d'attente répartie. Comparez le nombre de définitions nécessaires pour configurer le même réseau qu'un cluster.



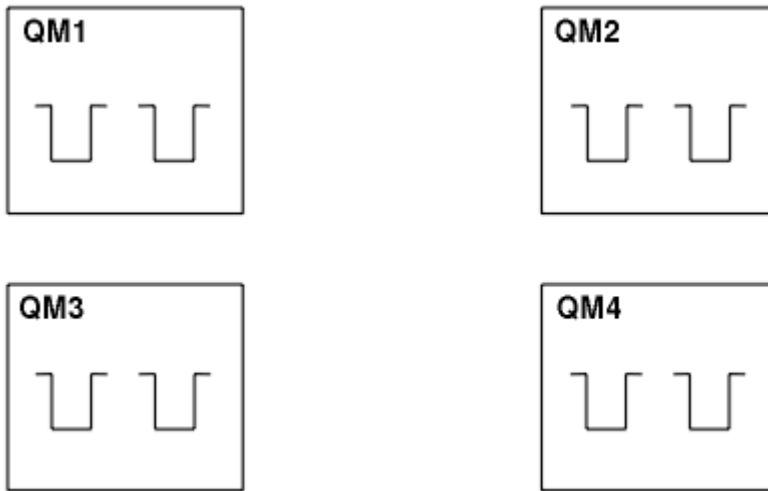


Figure 4. Un réseau de quatre gestionnaires de files d'attente

## Définitions permettant de configurer un réseau à l'aide de la mise en file d'attente répartie

Pour configurer le réseau affiché dans la Figure 3, à la page 32 à l'aide de la mise en file d'attente répartie, vous pouvez avoir les définitions suivantes:

Tableau 2. Définitions pour la mise en file d'attente répartie

Description	Nombre par gestionnaire de files d'attente	Nombre total
Une définition de canal émetteur pour un canal sur lequel envoyer des messages à tous les autres gestionnaires de files d'attente	3	12
Une définition de canal récepteur pour un canal sur lequel recevoir des messages d'un autre gestionnaire de files d'attente	3	12
Une définition de file d'attente de transmission pour une file d'attente de transmission vers tous les autres gestionnaires de files d'attente	3	12
Une définition de file d'attente locale pour chaque file d'attente locale	2	8
Une définition de file d'attente éloignée pour chaque file d'attente éloignée dans laquelle ce gestionnaire de files d'attente souhaite insérer des messages	6	24

Vous pouvez réduire ce nombre de définitions en utilisant des définitions de canal récepteur génériques. Le nombre maximal de définitions peut atteindre 17 sur chaque gestionnaire de files d'attente, soit un total de 68 pour ce réseau.

## Définitions pour la configuration d'un réseau à l'aide de clusters

Pour configurer le réseau affiché dans Figure 3, à la page 32 à l'aide de clusters, vous avez besoin des définitions suivantes:

Tableau 3. Définitions pour la mise en cluster

Description	Nombre par gestionnaire de files d'attente	Nombre total
Définition de canal émetteur de cluster pour un canal sur lequel envoyer des messages à un gestionnaire de files d'attente de référentiel	1	4
Une définition de canal récepteur de cluster pour un canal sur lequel recevoir des messages d'autres gestionnaires de files d'attente du cluster	1	4
Une définition de file d'attente locale pour chaque file d'attente locale	2	8

Pour configurer ce cluster de gestionnaires de files d'attente (avec deux référentiels complets), vous avez besoin de quatre définitions sur chaque gestionnaire de files d'attente, soit un total de seize définitions. Vous devez également modifier les définitions de gestionnaire de files d'attente pour deux des gestionnaires de files d'attente afin d'en faire des gestionnaires de files d'attente de référentiel complet pour le cluster.

Une seule définition de canal CLUSSDR et une seule définition de canal CLUSRCVR sont requises. Lorsque le cluster est défini, vous pouvez ajouter ou supprimer des gestionnaires de files d'attente (autres que les gestionnaires de files d'attente de référentiel) sans interruption pour les autres gestionnaires de files d'attente.

L'utilisation d'un cluster réduit le nombre de définitions requises pour configurer un réseau contenant de nombreux gestionnaires de files d'attente.

Avec moins de définitions à faire, il y a moins de risque d'erreur:

- Les noms d'objet correspondent toujours, par exemple le nom de canal dans une paire émetteur-récepteur.
- Le nom de file d'attente de transmission indiqué dans une définition de canal correspond toujours à la définition de file d'attente de transmission correcte ou au nom de file d'attente de transmission indiqué dans une définition de file d'attente éloignée.
- Une définition QREMOTE pointe toujours vers la file d'attente correcte sur le gestionnaire de files d'attente éloignées.

Une fois qu'un cluster est configuré, vous pouvez déplacer des files d'attente de cluster d'un gestionnaire de files d'attente à un autre au sein du cluster sans avoir à effectuer de travail de gestion système sur un autre gestionnaire de files d'attente. Il n'est pas possible d'oublier de supprimer ou de modifier des définitions de canal, de file d'attente éloignée ou de file d'attente de transmission. Vous pouvez ajouter de nouveaux gestionnaires de files d'attente à un cluster sans interruption du réseau existant.

### **Comment choisir des gestionnaires de files d'attente de cluster pour stocker des référentiels complets**

Dans chaque cluster, vous devez choisir au moins un, et de préférence deux, gestionnaires de files d'attente devant contenir des référentiels complets. Deux référentiels complets sont suffisants pour toutes les circonstances sauf les plus exceptionnelles. Si possible, choisissez des gestionnaires de files d'attente hébergés sur des plateformes robustes et connectées en permanence, qui ne coïncident pas avec des indisponibilités et qui se trouvent géographiquement en position centrale. Pensez également à dédier des systèmes en tant qu'hôtes de référentiel complets et à ne pas utiliser ces systèmes pour d'autres tâches.

Les *référentiels complets* sont des gestionnaires de files d'attente qui conservent une image complète de l'état du cluster. Pour partager ces informations, chaque référentiel complet est connecté par les canaux

CLUSSDR (et leurs définitions CLUSRCVR correspondantes) à tous les autres référentiels complets du cluster. Vous devez définir manuellement ces canaux.

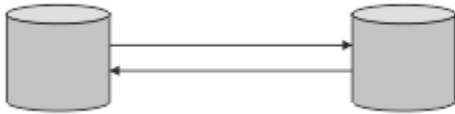


Figure 5. Deux référentiels complets connectés.

Tous les autres gestionnaires de files d'attente du cluster conservent une image de ce qu'ils connaissent actuellement sur l'état du cluster dans un *référentiel partiel*. Ces gestionnaires de files d'attente publient des informations sur eux-mêmes et demandent des informations sur les autres gestionnaires de files d'attente, à l'aide de deux référentiels complets disponibles. Si un référentiel complet choisi n'est pas disponible, un autre référentiel est utilisé. Lorsque le référentiel complet choisi redevient disponible, il collecte les informations nouvelles et modifiées les plus récentes auprès des autres afin qu'elles restent à l'étape. Si tous les référentiels complets sont mis hors service, les autres gestionnaires de files d'attente utilisent les informations dont ils disposent dans leurs référentiels partiels. Cependant, ils sont limités à l'utilisation des informations dont ils disposent ; les nouvelles informations et les demandes de mises à jour ne peuvent pas être traitées. Lorsque les référentiels complets se reconnectent au réseau, des messages sont échangés pour mettre à jour tous les référentiels (complets et partiels).

Lors de la planification de l'allocation de référentiels complets, tenez compte des points suivants:

- Les gestionnaires de files d'attente choisis pour contenir des référentiels complets doivent être fiables et gérés. Choisissez des gestionnaires de files d'attente hébergés sur une plateforme robuste et connectée en permanence.
- Prenez en compte les indisponibilités planifiées pour les systèmes hébergeant vos référentiels complets et assurez-vous qu'elles ne coïncident pas.
- Prenez en compte les performances du réseau: choisissez les gestionnaires de files d'attente qui se trouvent dans une position géographique centrale ou qui partagent le même système que les autres gestionnaires de files d'attente du cluster.
- Déterminez si un gestionnaire de files d'attente est membre de plusieurs clusters. Il peut être pratique sur le plan administratif d'utiliser le même gestionnaire de files d'attente pour héberger les référentiels complets de plusieurs clusters, à condition que cet avantage soit équilibré par rapport à l'occupation attendue du gestionnaire de files d'attente.
- Pensez à dédier certains systèmes pour qu'ils contiennent uniquement des référentiels complets et à ne pas utiliser ces systèmes pour d'autres tâches. Cela permet de s'assurer que ces systèmes ne nécessitent que la maintenance pour la configuration du gestionnaire de files d'attente et qu'ils ne sont pas retirés du service pour la maintenance d'autres applications métier. Il garantit également que la tâche de gestion du référentiel ne soit pas en concurrence avec les applications pour les ressources système. Cela peut être particulièrement utile dans les grands clusters (par exemple, les clusters de plus d'un millier de gestionnaires de files d'attente), où les référentiels complets ont une charge de travail beaucoup plus élevée pour la gestion de l'état du cluster.

Il est possible d'avoir plus de deux référentiels complets, mais rarement recommandé. Bien que les définitions d'objet (c'est-à-dire les files d'attente, les rubriques et les canaux) soient transmises à tous les référentiels complets disponibles, les demandes ne circulent que d'un référentiel partiel à un maximum de deux référentiels complets. Cela signifie que, lorsque plus de deux référentiels complets sont définis et que deux référentiels complets deviennent indisponibles, il se peut que certains référentiels partiels ne reçoivent pas les mises à jour qu'ils attendent. Voir [MQ Clusters: Pourquoi seulement deux référentiels complets?](#)

Une situation dans laquelle il peut s'avérer utile de définir plus de deux référentiels complets est la migration de référentiels complets existants vers de nouveaux matériels ou de nouveaux gestionnaires de files d'attente. Dans ce cas, vous devez introduire les référentiels complets de remplacement et confirmer qu'ils sont devenus complets, avant de supprimer les référentiels complets précédents. Chaque fois que

vous ajoutez un référentiel complet, n'oubliez pas que vous devez le connecter directement à tous les autres référentiels complets avec les canaux CLUSSDR .

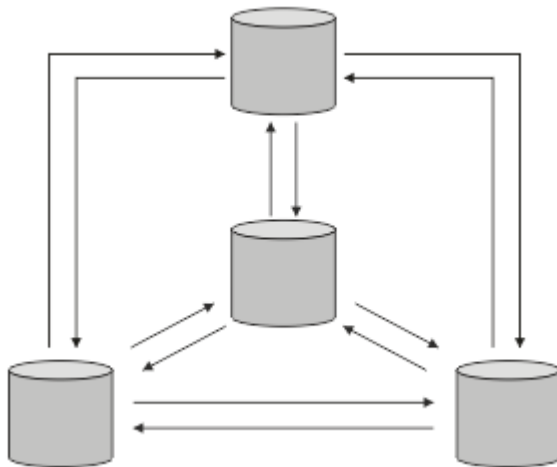


Figure 6. Plus de deux référentiels complets connectés

### Information associée

Clusters MQ : Pourquoi deux référentiels complets seulement?

Quelle est la taille d'un cluster MQ ?

### Organisation d'un cluster

Sélectionnez les gestionnaires de files d'attente à lier à un référentiel complet. Tenez compte de l'effet sur les performances, de la version du gestionnaire de files d'attente et déterminez si plusieurs canaux CLUSSDR sont souhaitables.

Après avoir sélectionné les gestionnaires de files d'attente devant contenir des référentiels complets, vous devez choisir les gestionnaires de files d'attente à lier à quel référentiel complet. La définition de canal CLUSSDR lie un gestionnaire de files d'attente à un référentiel complet à partir duquel il découvre les autres référentiels complets du cluster. A partir de ce moment, le gestionnaire de files d'attente envoie des messages à deux référentiels complets. Il tente toujours d'utiliser d'abord celle à laquelle il a une définition de canal CLUSSDR . Vous pouvez choisir de lier un gestionnaire de files d'attente à un référentiel complet. Lors du choix, tenez compte de la topologie de votre configuration et de l'emplacement physique ou géographique des gestionnaires de files d'attente.

Etant donné que toutes les informations de cluster sont envoyées à deux référentiels complets, il peut arriver que vous souhaitiez créer une deuxième définition de canal CLUSSDR . Vous pouvez définir un second canal CLUSSDR dans un cluster comportant de nombreux référentiels complets répartis sur une zone étendue. Vous pouvez ensuite contrôler les deux référentiels complets auxquels vos informations sont envoyées.

### Conventions de dénomination de cluster

Envisagez de nommer les gestionnaires de files d'attente dans le même cluster à l'aide d'une convention de dénomination qui identifie le cluster auquel appartient le gestionnaire de files d'attente. Utilisez une convention de dénomination similaire pour les noms de canal et étendez cette convention pour décrire les caractéristiques du canal.

### Meilleures pratiques lors de la désignation de clusters MQ

Bien que les noms de cluster puissent comporter jusqu'à 48 caractères, des noms de cluster relativement courts sont utiles lors de l'application de conventions de dénomination à d'autres objets. Voir «Meilleures pratiques lors de la sélection des noms de canal de cluster», à la page 37.

Lorsque vous choisissez un nom de cluster, il est généralement utile de représenter l'"objectif" du cluster (qui est susceptible d'être de longue durée de vie) plutôt que le "contenu". Par exemple, 'B2BPROD' ou 'ACTTEST' plutôt que 'QM1\_QM2\_QM3\_CLUS'.

## Meilleures pratiques lors de la sélection des noms de gestionnaire de files d'attente de cluster

Si vous créez un nouveau cluster et ses membres à partir de zéro, envisagez une convention de dénomination pour les gestionnaires de files d'attente qui reflète leur utilisation du cluster. Chaque gestionnaire de files d'attente doit avoir un nom différent. Toutefois, vous pouvez attribuer aux gestionnaires de files d'attente d'un cluster un ensemble de noms similaires pour faciliter l'identification et la mémorisation des regroupements logiques (par exemple, 'ACTTQM1, ACTTQM2).

Les noms de gestionnaire de files d'attente relativement courts (par exemple moins de 8 caractères) vous aident si vous choisissez d'utiliser la convention décrite dans la section suivante, ou quelque chose de similaire, pour les noms de canal.

## Meilleures pratiques lors de la sélection des noms de canal de cluster

Etant donné que les noms des gestionnaires de files d'attente et des clusters peuvent comporter jusqu'à 48 caractères et qu'un nom de canal est limité à 20 caractères, prenez soin de ne pas avoir à modifier la convention de dénomination au milieu d'un projet (voir la section précédente).

Lors de la définition des canaux, n'oubliez pas que les canaux émetteurs de cluster créés automatiquement sur n'importe quel gestionnaire de files d'attente du cluster prennent leur nom du canal récepteur de cluster correspondant configuré sur le gestionnaire de files d'attente récepteur du cluster, et que ces canaux doivent donc être uniques et avoir un sens *sur les gestionnaires de files d'attente éloignées du cluster*.

Une approche commune consiste à utiliser le nom de gestionnaire de files d'attente précédé du nom de cluster. Par exemple, si le nom de cluster est CLUSTER1 et que les gestionnaires de files d'attente sont QM1, QM2, les canaux récepteurs de cluster sont CLUSTER1.QM1, CLUSTER1.QM2.

Vous pouvez étendre cette convention si les canaux ont des priorités différentes ou utilisent des protocoles différents. Exemple :

- CLUSTER1.QM1.S1
- CLUSTER1.QM1.N3
- CLUSTER1.QM1.T4

Dans cet exemple, S1 peut être le premier canal SNA, N3 peut être le canal NetBIOS avec une priorité de réseau de trois et T4 peut être une adresse IP TCP utilisant un réseau IPV4 .

### Désignation des définitions de canal partagé

Une définition de canal unique peut être partagée entre plusieurs clusters, auquel cas les conventions de dénomination suggérées ici doivent être modifiées. Toutefois, comme décrit dans [Gestion des définitions de canal](#) , il est généralement préférable de définir des canaux discrets pour chaque cluster dans tous les cas.

### Anciennes conventions de dénomination des canaux

En dehors des environnements de cluster, il a toujours été courant d'utiliser une convention de dénomination 'FROMQM.TO.TARGETQM', de sorte que vous pouvez trouver des clusters existants qui ont utilisé quelque chose de similaire (par exemple, CLUSTER.TO.TARGET). Cela n'est pas recommandé dans le cadre d'un nouveau schéma de dénomination de cluster car il réduit davantage les caractères disponibles pour transmettre des informations "utiles" dans votre nom de canal.

### Noms de canal sur IBM MQ for z/OS

Vous pouvez définir des ressources génériques VTAM ou des noms génériques DDNS (*Dynamic Domain Name Server*). Vous pouvez définir des noms de connexion à l'aide de noms génériques.

Toutefois, lorsque vous créez une définition de récepteur de cluster, n'utilisez pas de nom de connexion générique.

Le problème lié à l'utilisation de noms de connexion génériques pour les définitions de récepteur de cluster est le suivant: si vous définissez un CLUSRCVR avec un CONNAME générique, il n'est pas garanti que vos canaux CLUSSDR pointent vers les gestionnaires de files d'attente que vous souhaitez. Votre CLUSSDR initial peut pointer vers n'importe quel gestionnaire de files d'attente du groupe de partage de files d'attente, pas nécessairement vers un gestionnaire qui héberge un référentiel complet. Si un canal recommence à établir une connexion, il peut se reconnecter à un gestionnaire de files d'attente différent portant le même nom générique, ce qui perturbe le flux de messages.

**z/OS**

### **Queue sharing groups and clusters**

Shared queues can be cluster queues and queue managers in a queue sharing group can also be cluster queue managers.

On IBM MQ for z/OS you can group queue managers into queue sharing groups. A queue manager in a queue sharing group can define a local queue that is to be shared by up to 32 queue managers.

Shared queues can also be cluster queues. Furthermore, the queue managers in a queue sharing group can also be in one or more clusters.

Vous pouvez définir des ressources génériques VTAM ou des noms génériques DDNS ( *Dynamic Domain Name Server* ). Vous pouvez définir des noms de connexion à l'aide de noms génériques. Toutefois, lorsque vous créez une définition de récepteur de cluster, n'utilisez pas de nom de connexion générique.

Le problème lié à l'utilisation de noms de connexion génériques pour les définitions de récepteur de cluster est le suivant: si vous définissez un CLUSRCVR avec un CONNAME générique, il n'est pas garanti que vos canaux CLUSSDR pointent vers les gestionnaires de files d'attente que vous souhaitez. Votre CLUSSDR initial peut pointer vers n'importe quel gestionnaire de files d'attente du groupe de partage de files d'attente, pas nécessairement vers un gestionnaire qui héberge un référentiel complet. Si un canal recommence à établir une connexion, il peut se reconnecter à un gestionnaire de files d'attente différent portant le même nom générique, ce qui perturbe le flux de messages.

A CLUSRCVR channel that uses the group listener port can not be started because, if this were the case, it would not be possible to tell which queue manager the CLUSRCVR would connect to each time. The cluster system queues on which information is kept about the cluster are not shared. Each queue manager has its own.

Cluster channels are used not only to transfer application messages but internal system messages about the setup of the cluster. Each queue manager in the cluster must receive these internal system messages to participate properly in clustering, so needs its own unique CLUSRCVR channel on which to receive them.

A shared CLUSRCVR could start on any queue manager in the queue sharing group (QSG) and so lead to an inconsistent supply of the internal system messages to the QSG queue managers, meaning none can properly participate in the cluster. To ensure no shared CLUSRCVR channels can be used, any attempt fails with the [CSQX502E](#) message.

### **Chevauchement de clusters**

Les clusters qui se chevauchent fournissent des fonctions d'administration supplémentaires. Utilisez des listes de noms pour réduire le nombre de commandes nécessaires à l'administration des clusters qui se chevauchent.

Vous pouvez créer des clusters qui se chevauchent. Il existe un certain nombre de raisons pour lesquelles vous pouvez définir des clusters qui se chevauchent, par exemple:

- Permettre à différentes organisations d'avoir leur propre administration.
- Permettre l'administration séparée des applications indépendantes.
- Pour créer des classes de service.

Dans Figure 7, à la page 39, le gestionnaire de files d'attente STF2 est membre des deux clusters. Lorsqu'un gestionnaire de files d'attente est membre de plusieurs clusters, vous pouvez tirer parti

des listes de noms pour réduire le nombre de définitions dont vous avez besoin. Les listes de noms contiennent une liste de noms, par exemple, des noms de cluster. Vous pouvez créer une liste de noms nommant les clusters. Spécifiez la liste de noms dans la commande ALTER QMGR pour STF2 afin d'en faire un gestionnaire de files d'attente de référentiel complet pour les deux clusters.

Si votre réseau comporte plusieurs clusters, vous devez leur attribuer des noms différents. Si deux clusters portant le même nom sont fusionnés, il n'est pas possible de les séparer à nouveau. Il est également judicieux de donner des noms différents aux clusters et aux canaux. Ils sont plus faciles à distinguer lorsque vous examinez la sortie des commandes DISPLAY. Les noms de gestionnaire de files d'attente doivent être uniques dans un cluster pour qu'il fonctionne correctement.

### Définition de classes de service

Imaginez une université qui a un gestionnaire de files d'attente pour chaque membre du personnel et chaque étudiant. Les messages entre les membres du personnel doivent circuler sur des canaux ayant une priorité élevée et une bande passante élevée. Les messages entre les étudiants doivent voyager sur des canaux moins chers et plus lents. Vous pouvez configurer ce réseau à l'aide des techniques traditionnelles de mise en file d'attente répartie. IBM MQ sélectionne les canaux à utiliser en examinant le nom de la file d'attente de destination et le nom du gestionnaire de files d'attente.

Pour différencier clairement le personnel et les étudiants, vous pouvez regrouper leurs gestionnaires de files d'attente dans deux clusters, comme illustré dans la [Figure 7](#), à la page 39. IBM MQ déplace les messages vers la file d'attente des réunions dans le cluster de personnel uniquement sur les canaux définis dans ce cluster. Les messages de la file d'attente gossip dans le cluster d'étudiants passent par les canaux définis dans ce cluster et reçoivent la classe de service appropriée.

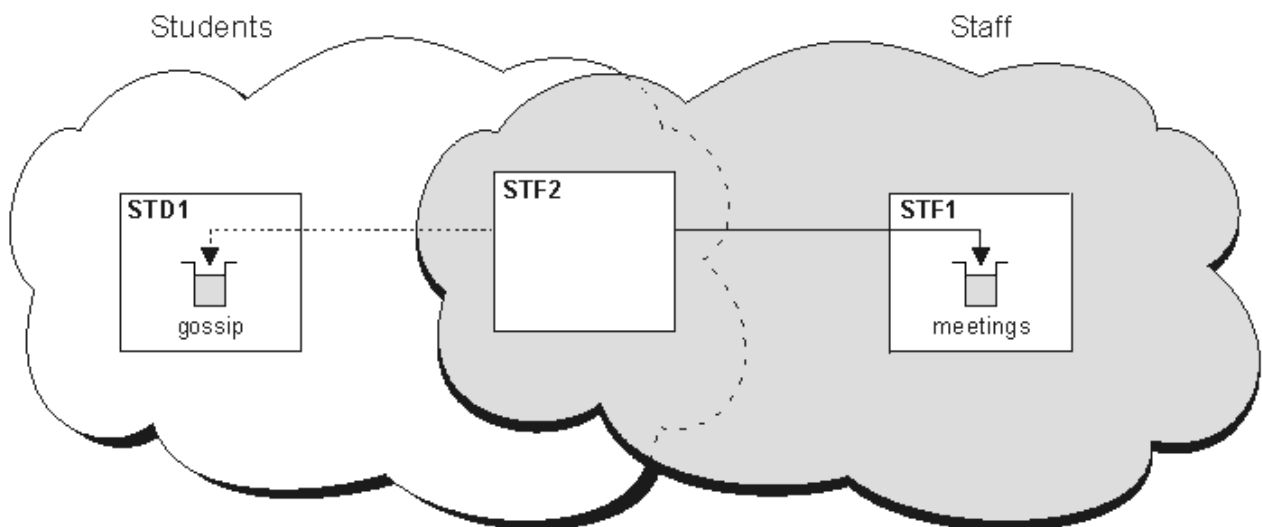


Figure 7. Classes de service

### Conseils de mise en cluster

Vous devrez peut-être apporter des modifications à vos systèmes ou à vos applications avant d'utiliser la mise en cluster. Il existe des similitudes et des différences par rapport au comportement de la mise en file d'attente répartie.

- Vous devez ajouter des définitions de configuration manuelle aux gestionnaires de files d'attente en dehors d'un cluster pour qu'ils puissent accéder aux files d'attente de cluster.
- Si vous fusionnez deux clusters portant le même nom, vous ne pouvez pas les séparer à nouveau. Par conséquent, il est conseillé d'attribuer un nom unique à tous les clusters.
- Si un message arrive dans un gestionnaire de files d'attente mais qu'il n'y a pas de file d'attente pour le recevoir, le message est placé dans la file d'attente de rebut. S'il n'existe pas de file d'attente de rebut, le canal échoue et effectue une nouvelle tentative. L'utilisation de la file d'attente de rebut est identique à celle de la file d'attente répartie.

- L'intégrité des messages persistants est maintenue. Les messages ne sont pas dupliqués ou perdus suite à l'utilisation de clusters.
- L'utilisation de clusters réduit l'administration du système. Les clusters facilitent la connexion de réseaux plus grands avec beaucoup plus de gestionnaires de files d'attente que vous ne pouvez envisager d'utiliser la mise en file d'attente répartie. Vous risquez de consommer un nombre excessif de ressources réseau si vous tentez d'activer la communication entre chaque gestionnaire de files d'attente d'un cluster.
- Si vous utilisez l' IBM MQ Explorer, qui présente les gestionnaires de files d'attente dans une structure arborescente, la vue des grands clusters peut être lourde.
- **Multi** L'objectif des listes de distribution est d'utiliser une seule commande MQPUT pour envoyer le même message à plusieurs destinations. Les listes de distribution sont prises en charge sur IBM MQ for Multiplatforms. Vous pouvez utiliser des listes de distribution avec des clusters de gestionnaires de files d'attente. Dans un cluster, tous les messages sont développés à l'heure MQPUT . L'avantage, en termes de trafic réseau, n'est pas aussi grand que dans un environnement non-cluster. L'avantage des listes de distribution est que les nombreux canaux et files d'attente de transmission n'ont pas besoin d'être définis manuellement.
- Si vous prévoyez d'utiliser des clusters pour équilibrer votre charge de travail, examinez vos applications. Vérifiez s'ils nécessitent que les messages soient traités par un gestionnaire de files d'attente particulier ou dans un ordre particulier. De telles applications sont dites avoir des affinités de messages. Vous devrez peut-être modifier vos applications avant de pouvoir les utiliser dans des clusters complexes.
- Vous pouvez choisir d'utiliser l'option MQOO\_BIND\_ON\_OPEN sur un MQOPEN pour forcer l'envoi de messages à une destination spécifique. Si le gestionnaire de files d'attente de destination n'est pas disponible, les messages ne sont pas distribués tant que le gestionnaire de files d'attente n'est pas à nouveau disponible. Les messages ne sont pas acheminés vers un autre gestionnaire de files d'attente en raison du risque de duplication.
- Si un gestionnaire de files d'attente doit héberger un référentiel de cluster, vous devez connaître son nom d'hôte ou son adresse IP. Vous devez spécifier ces informations dans le paramètre CONNAME lorsque vous créez la définition CLUSSDR sur d'autres gestionnaires de files d'attente qui rejoignent le cluster. Si vous utilisez DHCP, l'adresse IP est susceptible d'être modifiée car DHCP peut allouer une nouvelle adresse IP chaque fois que vous redémarrez un système. Par conséquent, vous ne devez pas spécifier l'adresse IP dans les définitions CLUSSDR . Même si toutes les définitions CLUSSDR spécifient le nom d'hôte plutôt que l'adresse IP, les définitions ne sont toujours pas fiables. DHCP ne met pas nécessairement à jour l'entrée de répertoire DNS pour l'hôte avec la nouvelle adresse. Si vous devez désigner des gestionnaires de files d'attente en tant que référentiels complets sur les systèmes qui utilisent DHCP, installez un logiciel garantissant la mise à jour de votre répertoire DNS.
- N'utilisez pas de noms génériques, par exemple des ressources génériques VTAM ou des noms génériques DDNS (Dynamic Domain Name Server) comme noms de connexion pour vos canaux. Dans ce cas, vos canaux peuvent se connecter à un gestionnaire de files d'attente différent de celui attendu.
- Vous pouvez uniquement extraire un message d'une file d'attente de cluster locale, mais vous pouvez insérer un message dans n'importe quelle file d'attente d'un cluster. Si vous ouvrez une file d'attente pour utiliser la commande MQGET , le gestionnaire de files d'attente ouvre la file d'attente locale.
- Vous n'avez pas besoin de modifier vos applications si vous configurez un cluster IBM MQ simple. L'application peut nommer la file d'attente cible sur l'appel MQOPEN et n'a pas besoin de connaître l'emplacement du gestionnaire de files d'attente. Si vous configurez un cluster pour la gestion de charge de travail, vous devez examiner vos applications et les modifier si nécessaire.
- Vous pouvez afficher les données de surveillance et de statut en cours d'un canal ou d'une file d'attente à l'aide des commandes DISPLAY CHSTATUS et DISPLAY QSTATUS **runmqsc** . Les informations de surveillance peuvent être utilisées pour évaluer les performances et la santé du système. La surveillance est contrôlée par des attributs de gestionnaire de files d'attente, de file d'attente et de canal. La surveillance des canaux émetteurs de cluster définis automatiquement est possible avec l'attribut de gestionnaire de files d'attente MONACLS .



## Concepts associés

### Clusters

«Comparaison de la mise en cluster et de la mise en file d'attente répartie», à la page 32

Comparez les composants qui doivent être définis pour connecter les gestionnaires de files d'attente à l'aide de la mise en file d'attente répartie et de la mise en cluster.

### Composants d'un cluster

## Tâches associées

Configuration d'un cluster de gestionnaires de files d'attente

Configuration d'un nouveau cluster

## **Combien de temps les référentiels de gestionnaire de files d'attente conservent-ils des informations ?**

Les référentiels de gestionnaire de files d'attente conservent les informations pendant 30 jours. Un processus automatique actualise efficacement les informations utilisées.

Lorsqu'un gestionnaire de files d'attente envoie des informations sur lui-même, les gestionnaires de files d'attente de référentiel complet et partiel stockent ces informations pendant 30 jours. Les informations sont envoyées, par exemple, lorsqu'un gestionnaire de files d'attente annonce la création d'une nouvelle file d'attente. Pour éviter que ces informations n'expirent, les gestionnaires de files d'attente se renvoient automatiquement après 27 jours. Si un référentiel partiel envoie une nouvelle demande d'informations au cours de la durée de vie de 30 jours, le délai d'expiration reste le délai d'origine de 30 jours.

Lorsque les informations arrivent à expiration, elles ne sont pas immédiatement supprimées du référentiel. Au lieu de cela, il est conservé pendant une période de grâce de 60 jours. Si aucune mise à jour n'est reçue dans le délai de grâce, les informations sont supprimées. Le délai de grâce tient compte du fait qu'un gestionnaire de files d'attente peut avoir été temporairement hors service à la date d'expiration. Si un gestionnaire de files d'attente est déconnecté d'un cluster pendant plus de 90 jours, il cesse de faire partie du cluster. Toutefois, s'il se reconnecte au réseau, il fait à nouveau partie du cluster. Les référentiels complets n'utilisent pas les informations arrivées à expiration pour répondre aux nouvelles demandes des autres gestionnaires de files d'attente.

De même, lorsqu'un gestionnaire de files d'attente envoie une demande d'informations à jour à partir d'un référentiel complet, la demande dure 30 jours. Après 27 jours, IBM MQ vérifie la demande. S'il a été référencé pendant les 27 jours, il est actualisé automatiquement. Si ce n'est pas le cas, il expire et est actualisé par le gestionnaire de files d'attente si nécessaire à nouveau. L'expiration des demandes empêche l'accumulation de demandes d'informations provenant des gestionnaires de files d'attente dormants.

**Remarque :** Vous devez télécharger et installer la PTF de l' APAR PH43191, qui corrige les erreurs système lors du calcul de l'heure d'expiration d'un abonnement. Ces erreurs peuvent entraîner une expiration anticipée de l'abonnement (entraînant l'émission du message CSQX456I ) ou une expiration après l'expiration de l'objet (entraînant des erreurs MQRC 2085 (MQRC\_UNKNOWN\_OBJECT)).

Pour les clusters de grande taille, il peut être perturbateur si de nombreux gestionnaires de files d'attente renvoient automatiquement toutes les informations les concernant en même temps. Voir L'actualisation d'un grand cluster peut affecter les performances et la disponibilité du cluster.

## Concepts associés

«Mise en cluster : meilleures pratiques d'utilisation REFRESH CLUSTER», à la page 75

Vous utilisez la commande **REFRESH CLUSTER** pour supprimer toutes les informations stockées localement sur un cluster et régénérer ces informations à partir des référentiels complets du cluster. Vous ne devez pas utiliser cette commande, sauf dans des circonstances exceptionnelles. Si vous avez besoin de l'utiliser, il y a des considérations spéciales sur la façon dont vous l'utilisez. Ces informations sont un guide basé sur les tests et les commentaires des clients.

## **Exemples de clusters**

Le premier exemple montre le plus petit cluster possible de deux gestionnaires de files d'attente. Les deuxième et troisième exemples présentent deux versions d'un cluster de trois gestionnaires de files d'attente.

Le plus petit cluster possible ne contient que deux gestionnaires de files d'attente. Dans ce cas, les deux gestionnaires de files d'attente contiennent des référentiels complets. Vous n'avez besoin que de quelques définitions pour configurer le cluster, mais il existe un degré élevé d'autonomie au niveau de chaque gestionnaire de files d'attente.

### DEMOCLSTR

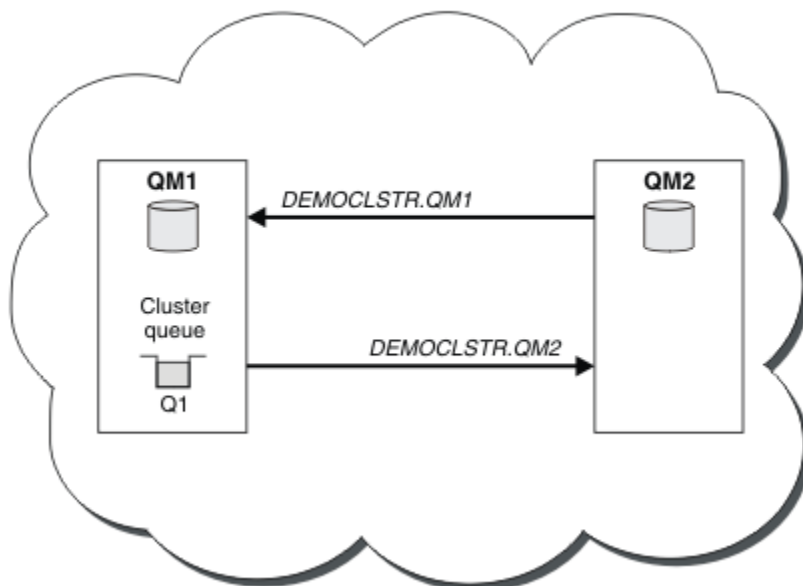


Figure 8. Un petit cluster de deux gestionnaires de files d'attente

- Les gestionnaires de files d'attente peuvent avoir des noms longs tels que LONDON et NEWYORK.  
z/OS Sur IBM MQ for z/OS, les noms de gestionnaire de files d'attente sont limités à quatre caractères.
- Chaque gestionnaire de files d'attente est généralement configuré sur une machine distincte. Toutefois, vous pouvez avoir plusieurs gestionnaires de files d'attente sur la même machine.

Pour obtenir des instructions sur la configuration d'un exemple de cluster similaire, voir [Configuration d'un nouveau cluster](#).

La Figure 9, à la page 43 montre les composants du cluster CLSTR1.

- Ce cluster contient trois gestionnaires de files d'attente, QM1, QM2 et QM3.
- QM1 et QM2 contiennent des référentiels d'informations sur tous les gestionnaires de files d'attente et les objets associés au cluster dans le cluster. Ces référentiels s'appellent des *gestionnaires de files d'attente de référentiel complet*. Les référentiels sont représentés dans le diagramme par les cylindres ombrés.
- QM2 et QM3 hébergent des files d'attente accessibles à n'importe quel autre gestionnaire de files d'attente dans le cluster. Les files d'attente accessibles à n'importe quel autre gestionnaire de files d'attente du cluster sont appelées *files d'attente de cluster*. Dans le diagramme, les files d'attente de cluster sont représentées par les files d'attente ombrées. Les files d'attente de cluster sont accessibles à partir de n'importe quel emplacement du cluster. Le code de mise en cluster IBM MQ garantit que les définitions de file d'attente éloignée des files d'attente du cluster sont créées dans n'importe quel gestionnaire de files d'attente qui y fait référence.

Comme dans le cas du groupement réparti, une application utilise l'appel MQPUT pour placer un message dans une file d'attente de cluster dans n'importe quel gestionnaire de files d'attente du cluster. Une application utilise l'appel MQGET pour extraire des messages d'une file d'attente de cluster dans le gestionnaire de files d'attente où se trouve la file d'attente.

- Chaque gestionnaire de files d'attente possède une définition créée manuellement pour l'extrémité réceptrice d'un canal appelé *cluster\_name*. *queue\_manager\_name* sur lequel il peut recevoir des messages. Sur le gestionnaire de files d'attente de réception, *cluster\_name.queue\_manager\_name* est un canal récepteur de cluster. Un canal de récepteur de cluster est identique à un canal récepteur utilisé dans le groupement réparti ; il reçoit les messages du gestionnaire de files d'attente. En outre, il reçoit les informations sur le cluster.

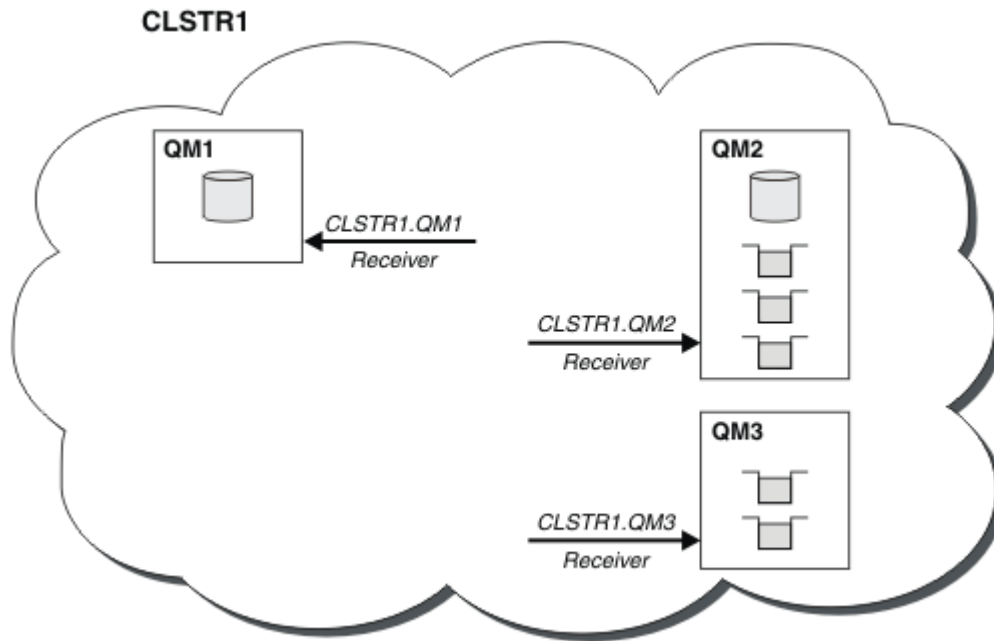


Figure 9. Cluster de gestionnaires de files d'attente

- Dans la Figure 10, à la page 44 chaque gestionnaire de files d'attente a une définition pour la partie émission d'un canal. Il se connecte au canal récepteur de files de cluster de l'un des gestionnaires de files d'attente de référentiel complet. Sur le gestionnaire de files d'attente émetteur, *cluster\_name.queue\_manager\_name* est un canal émetteur de cluster. QM1 et QM3 ont des canaux émetteurs de cluster qui se connectent à CLSTR1.QM2 ; voir la ligne pointillée "2".

QM2 a un canal émetteur qui se connecte à CLSTR1.QM1, voir la ligne pointillée "3". Un canal émetteur de cluster est identique à un canal émetteur utilisé dans la mise en file d'attente répartie ; il envoie des messages au gestionnaire de files d'attente de réception. En outre il envoie les informations sur le cluster.

Une fois que l'extrémité réceptrice de cluster et l'extrémité émettrice de cluster d'un canal sont définies, le canal démarre automatiquement.

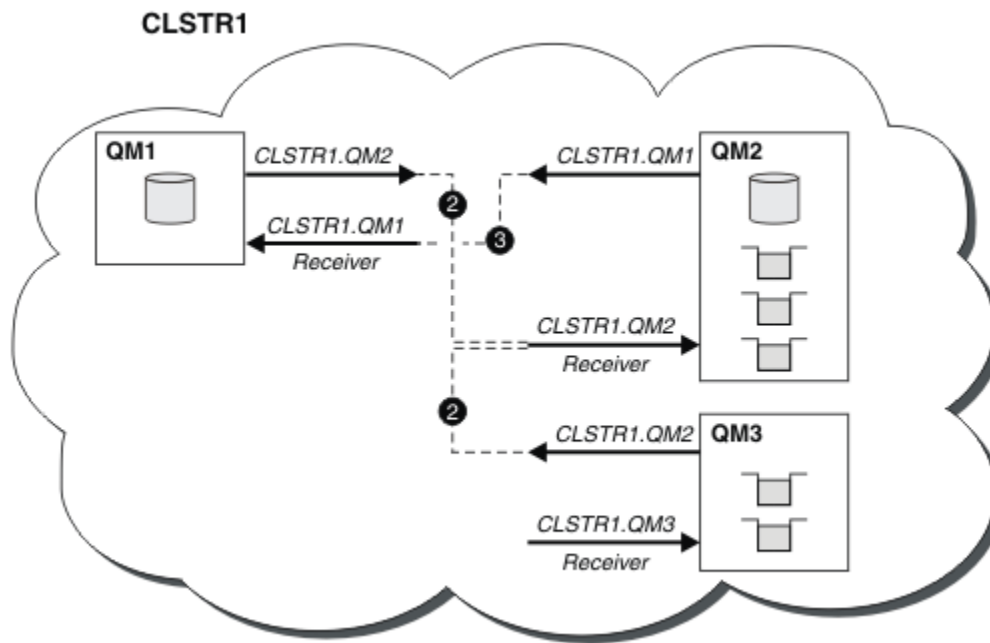


Figure 10. Cluster de gestionnaires de files d'attente avec des canaux émetteurs

La définition d'un canal émetteur de cluster dans le gestionnaire de files d'attente local présente ce gestionnaire de files d'attente à l'un des gestionnaires de files d'attente de référentiel complet. Ce dernier met à jour les informations dans son référentiel complet en conséquence. Ensuite, il crée automatiquement un canal émetteur de cluster dans le gestionnaire de files d'attente d'origine et envoie ces informations de gestionnaire de files d'attente sur le cluster. Un gestionnaire de files d'attente découvre alors un cluster et un cluster découvre un gestionnaire de files d'attente.

Reportez-vous de nouveau à la Figure 9, à la page 43. Supposons qu'une application connectée au gestionnaire de files d'attente QM3 veuille envoyer des messages aux files d'attente sur QM2. La première fois que QM3 doit accéder à ces files d'attente, il les reconnaît en consultant un référentiel complet. Dans ce cas, le référentiel complet est QM2 qui est accessible par le biais du canal émetteur CLSTR1.QM2. Avec les informations provenant de ce référentiel, il peut créer automatiquement des définitions éloignées pour ces files d'attente. Si les files d'attente se trouvent sur QM1, ce mécanisme fonctionne aussi car QM2 et un référentiel complet. Ce dernier comporte un enregistrement complet de tous les objets du cluster. Dans ce dernier cas, QM3 crée également un canal émetteur de cluster correspondant au canal récepteur de cluster sur QM1 pour leur permettre de communiquer.

La Figure 11, à la page 45 présente le même cluster pour lequel deux canaux émetteurs de cluster ont été créés automatiquement. Les canaux émetteurs de cluster sont représentés par les deux lignes pointillées qui joignent le canal récepteur de cluster CLSTR1.QM3. Il montre également la file d'attente de transmission de cluster, SYSTEM.CLUSTER.TRANSMIT.QUEUE, qu'utilise QM1 pour envoyer ses messages. Tous les gestionnaires de files d'attente dans le cluster disposent d'une file d'attente de transmission de cluster depuis laquelle ils peuvent envoyer des messages à n'importe quel autre gestionnaire de files d'attente dans le même cluster.

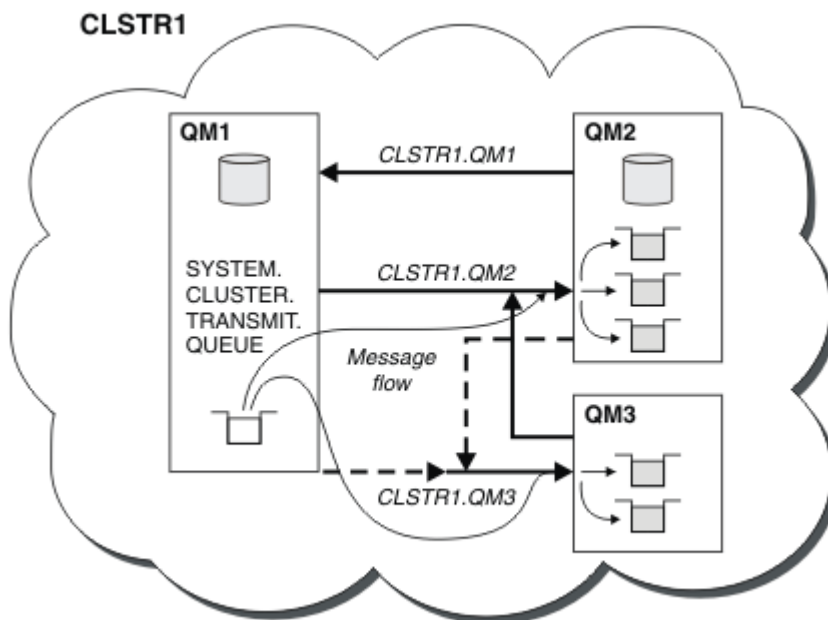


Figure 11. Cluster de gestionnaires de files d'attente montrant des canaux définis automatiquement

**Remarque :** Les autres diagrammes montrent uniquement les extrémités réceptrices des canaux pour lesquelles vous avez créé des définitions manuellement. Les extrémités émettrices sont omises, car elles sont généralement définies automatiquement, si nécessaire. La définition automatique de la plupart des canaux émetteur de cluster est essentielle au fonctionnement et à l'efficacité des clusters.

#### Concepts associés

«Comparaison de la mise en cluster et de la mise en file d'attente répartie», à la page 32

Comparez les composants qui doivent être définis pour connecter les gestionnaires de files d'attente à l'aide de la mise en file d'attente répartie et de la mise en cluster.

[Composants d'un cluster](#)

#### Tâches associées

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

[Configuration d'un nouveau cluster](#)

#### Mise en cluster: meilleures pratiques

Les clusters fournissent un mécanisme d'interconnexion des gestionnaires de files d'attente. Les meilleures pratiques décrites dans cette section sont basées sur les tests et les commentaires des clients.

Une configuration de cluster réussie dépend d'une bonne planification et d'une compréhension approfondie des principes de base de IBM MQ, tels que la bonne gestion des applications et la conception du réseau. Assurez-vous de bien connaître les informations des rubriques connexes avant de continuer.

#### Concepts associés

[Mise en file d'attente répartie et clusters](#)

[Clusters](#)

#### Tâches associées

«Conception des clusters», à la page 25

Les clusters fournissent un mécanisme d'interconnexion des gestionnaires de files d'attente qui simplifie à la fois la configuration initiale et la gestion en cours. Les clusters doivent être soigneusement conçus

pour s'assurer qu'ils fonctionnent correctement et qu'ils atteignent les niveaux de disponibilité et de réactivité requis.

### Surveillance des clusters

#### *Mise en cluster: considérations spéciales pour le chevauchement de clusters*

Cette rubrique fournit des conseils pour la planification et l'administration des clusters IBM MQ . Ces informations sont un guide basé sur les tests et les commentaires des clients.

## **Propriété de cluster**

Familiarisez-vous avec les clusters qui se chevauchent avant de lire les informations suivantes. Pour obtenir les informations nécessaires, voir «Chevauchement de clusters», à la page 38 et Configuration des chemins d'accès aux messages entre les clusters .

Lors de la configuration et de la gestion d'un système composé de clusters qui se chevauchent, il est préférable de respecter les points suivants:

- Bien que les clusters IBM MQ soient "faiblement couplés" comme décrit précédemment, il est utile de considérer un cluster comme une seule unité d'administration. Ce concept est utilisé car l'interaction entre les définitions sur des gestionnaires de files d'attente individuels est essentielle au bon fonctionnement du cluster. Par exemple, lorsque vous utilisez des files d'attente de cluster dont la charge de travail est équilibrée, il est important qu'un seul administrateur ou une seule équipe comprenne l'ensemble complet des destinations possibles pour les messages, qui dépend des définitions réparties dans le cluster. Plus trivialement, les paires de canaux émetteur / récepteur de cluster doivent être compatibles partout.
- Compte tenu de ce concept précédent, lorsque plusieurs clusters se rencontrent (qui doivent être administrés par des équipes / individus distincts), il est important de mettre en place des stratégies claires contrôlant l'administration des gestionnaires de files d'attente de passerelle.
- Il est utile de traiter les clusters qui se chevauchent comme un espace de nom unique: les noms de canal et les noms de gestionnaire de files d'attente doivent être uniques dans un cluster unique. L'administration est beaucoup plus facile lorsqu'elle est unique dans toute la topologie. Il est préférable de suivre une convention de dénomination appropriée. Les conventions possibles sont décrites dans «Conventions de dénomination de cluster», à la page 36.
- Parfois, la coopération en matière d'administration et de gestion des systèmes est essentielle. Par exemple, la coopération entre les organisations qui possèdent différents clusters qui doivent se chevaucher. Une compréhension claire des propriétaires de quoi et des règles et conventions applicables permet au groupement de s'exécuter sans heurt en cas de chevauchement de clusters.

## **Chevauchement de clusters: passerelles**

En général, un cluster unique est plus facile à administrer que plusieurs clusters. Par conséquent, la création d'un grand nombre de petits clusters (un pour chaque application par exemple) est généralement à éviter.

Toutefois, pour fournir des classes de service, vous pouvez implémenter des clusters qui se chevauchent. Exemple :

- Si vous avez des clusters concentriques dans lesquels le plus petit est destiné à la publication / abonnement. Pour plus d'informations, voir Comment dimensionner les systèmes .
- Si certains gestionnaires de files d'attente doivent être administrés par des équipes différentes. Pour plus d'informations, voir la section précédente «Propriété de cluster», à la page 46.
- Si cela a un sens d'un point de vue organisationnel ou géographique.
- Si des clusters équivalents fonctionnent avec la résolution de nom, par exemple lors de l'implémentation de TLS dans un cluster existant.

Les clusters qui se chevauchent ne présentent aucun avantage en matière de sécurité ; ils permettent aux clusters administrés par deux équipes différentes de se chevaucher et de rejoindre efficacement les équipes ainsi que la topologie:

- Tout nom annoncé dans un tel cluster est accessible à l'autre cluster.
- Tout nom annoncé dans un cluster peut être annoncé dans l'autre pour retirer des messages éligibles.
- Tout objet non annoncé sur un gestionnaire de files d'attente adjacent à la passerelle peut être résolu à partir de tout cluster dont la passerelle est membre.

L'espace de nom est l'union des deux clusters et doit être traité comme un espace de nom unique. Par conséquent, la propriété d'un cluster qui se chevauche est partagée entre tous les administrateurs des deux clusters.

Lorsqu'un système contient plusieurs clusters, il peut être nécessaire d'acheminer les messages des gestionnaires de files d'attente d'un cluster vers les files d'attente des gestionnaires de files d'attente d'un autre cluster. Dans cette situation, les clusters multiples doivent être interconnectés d'une manière ou d'une autre: l'utilisation de gestionnaires de files d'attente de passerelle entre les clusters est un bon modèle à suivre. Cette disposition évite de créer un maillage de canaux point à point difficile à gérer et fournit un bon endroit pour gérer des problèmes tels que les politiques de sécurité. Il existe deux manières distinctes de parvenir à cet accord:

1. Placez un ou plusieurs gestionnaires de files d'attente dans les deux clusters à l'aide d'une deuxième définition de récepteur de cluster. Cet arrangement implique moins de définitions administratives mais, comme indiqué précédemment, signifie que la propriété d'un cluster qui se chevauche est partagée entre tous les administrateurs des deux clusters.
2. Appariez un gestionnaire de files d'attente dans le cluster un avec un gestionnaire de files d'attente dans le cluster deux à l'aide de canaux point-à-point traditionnels.

Dans l'un ou l'autre de ces cas, divers outils peuvent être utilisés pour acheminer le trafic de manière appropriée. En particulier, les alias de file d'attente ou de gestionnaire de files d'attente peuvent être utilisés pour l'acheminement vers l'autre cluster, et un alias de gestionnaire de files d'attente avec une propriété **QMNAME** vide redirige l'équilibrage de charge là où il est souhaité.

### Concepts associés

[«Conventions de dénomination de cluster», à la page 36](#)

Envisagez de nommer les gestionnaires de files d'attente dans le même cluster à l'aide d'une convention de dénomination qui identifie le cluster auquel appartient le gestionnaire de files d'attente. Utilisez une convention de dénomination similaire pour les noms de canal et étendez cette convention pour décrire les caractéristiques du canal.

*Mise en cluster: remarques sur la conception de la topologie*

Cette rubrique fournit des conseils pour la planification et l'administration des clusters IBM MQ . Ces informations sont un guide basé sur les tests et les commentaires des clients.

En réfléchissant à l'endroit où les applications utilisateur et les processus administratifs internes seront localisés à l'avance, de nombreux problèmes peuvent être évités ou minimisés à une date ultérieure. Cette rubrique contient des informations sur les décisions de conception qui peuvent améliorer les performances et simplifier les tâches de maintenance à mesure que le cluster évolue.

- [«Performances de l'infrastructure de mise en cluster», à la page 47](#)
- [«Référentiels complets», à la page 48](#)
- [«Les applications devraient-elles utiliser des files d'attente dans les référentiels complets?», à la page 49](#)
- [«Gestion des définitions de canal», à la page 50](#)
- [«Equilibrage de la charge de travail sur plusieurs canaux», à la page 50](#)

### Performances de l'infrastructure de mise en cluster

Lorsqu'une application tente d'ouvrir une file d'attente sur un gestionnaire de files d'attente dans un cluster, le gestionnaire de files d'attente enregistre son intérêt auprès des référentiels complets pour cette file d'attente afin de savoir où se trouve la file d'attente dans le cluster. Toute mise à jour de l'emplacement ou de la configuration de la file d'attente est automatiquement envoyée par les référentiels complets au gestionnaire de files d'attente concerné. Cet enregistrement d'intérêt est connu

en interne sous le nom d'abonnement (ces abonnements ne sont pas les mêmes que les abonnements IBM MQ utilisés pour la messagerie de publication / abonnement dans IBM MQ )

Toutes les informations sur un cluster passent par chaque référentiel complet. Les référentiels complets sont donc toujours utilisés dans un cluster pour le trafic des messages d'administration. L'utilisation élevée des ressources système lors de la gestion de ces abonnements, ainsi que leur transmission et les messages de configuration qui en résultent, peuvent entraîner une charge considérable sur l'infrastructure de mise en cluster. Il y a un certain nombre de choses à prendre en compte pour s'assurer que cette charge est comprise et réduite dans la mesure du possible:

- Plus le nombre de gestionnaires de files d'attente individuels utilisant une file d'attente de cluster est élevé, plus le nombre d'abonnements dans le système est élevé, ce qui augmente le temps système d'administration lorsque des modifications sont apportées et que les abonnés intéressés doivent être avertis, en particulier sur les gestionnaires de files d'attente de référentiel complet. Une façon de réduire le trafic inutile et la charge de référentiel complète consiste à connecter des applications similaires (c'est-à-dire celles qui fonctionnent avec les mêmes files d'attente) à un plus petit nombre de gestionnaires de files d'attente.
- Outre le nombre d'abonnements dans le système qui affecte les performances, le taux de modification de la configuration des objets en cluster peut affecter les performances, par exemple le changement fréquent d'une configuration de file d'attente en cluster.
- Lorsqu'un gestionnaire de files d'attente est membre de plusieurs clusters (c'est-à-dire qu'il fait partie d'un système de cluster qui se chevauche), tout intérêt dans une file d'attente se traduit par un abonnement pour chaque cluster dont il est membre, même si les mêmes gestionnaires de files d'attente sont les référentiels complets de plusieurs clusters. Cette disposition augmente la charge sur le système et permet de déterminer si plusieurs clusters qui se chevauchent sont nécessaires plutôt qu'un seul cluster.
- Le trafic des messages d'application (c'est-à-dire les messages envoyés par les applications IBM MQ aux files d'attente de cluster) ne passe pas par les référentiels complets pour atteindre les gestionnaires de files d'attente de destination. Ce trafic de messages est envoyé directement entre le gestionnaire de files d'attente dans lequel le message entre dans le cluster et le gestionnaire de files d'attente dans lequel se trouve la file d'attente de cluster. Il n'est donc pas nécessaire de prendre en charge des débits élevés de trafic de messages d'application par rapport aux gestionnaires de files d'attente de référentiel complet, sauf si les gestionnaires de files d'attente de référentiel complet sont l'un des deux gestionnaires de files d'attente mentionnés. Pour cette raison, il est recommandé que les gestionnaires de files d'attente de référentiel complet ne soient pas utilisés pour le trafic des messages d'application dans les clusters où la charge de l'infrastructure de mise en cluster est importante.

## Référentiels complets

Un référentiel est un ensemble d'informations concernant les gestionnaires de files d'attente membres d'un cluster. 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. Pour plus d'informations sur les référentiels complets et les référentiels partiels, voir [Référentiel de cluster](#).

Les référentiels complets doivent être conservés sur des serveurs fiables et aussi hautement disponibles que possible et les points de défaillance uniques doivent être évités. La conception du cluster doit toujours comporter deux référentiels complets. En cas d'échec d'un référentiel complet, le cluster peut toujours fonctionner.

Les détails des mises à jour apportées aux ressources de cluster par un gestionnaire de files d'attente dans un cluster, par exemple les files d'attente en cluster, sont envoyés de ce gestionnaire de files d'attente à deux référentiels complets au maximum dans ce cluster (ou à un seul s'il n'existe qu'un seul gestionnaire de files d'attente de référentiel complet dans le cluster). Ces référentiels complets contiennent les informations et les propagent à tous les gestionnaires de files d'attente du cluster qui s'y intéressent (c'est-à-dire qu'ils s'y abonnent). Pour vous assurer que chaque membre du cluster dispose d'une vue à jour des ressources du cluster, chaque gestionnaire de files d'attente doit pouvoir communiquer avec au moins un gestionnaire de files d'attente de référentiel complet à la fois.



Si, pour une raison quelconque, un gestionnaire de files d'attente ne peut pas communiquer avec des référentiels complets, il peut continuer à fonctionner dans le cluster en fonction de son niveau d'informations déjà mis en cache pendant un certain temps, mais aucune nouvelle mise à jour ou accès aux ressources de cluster précédemment inutilisées n'est disponible.

Pour cette raison, vous devez vous efforcer de garder les deux référentiels complets disponibles à tout moment. Toutefois, cet arrangement ne signifie pas que des mesures extrêmes doivent être prises parce que le cluster fonctionne de manière adéquate pendant une courte période sans dépôt complet.

Il existe une autre raison pour laquelle un cluster doit avoir deux gestionnaires de files d'attente de référentiel complet, autre que la disponibilité des informations de cluster: cette raison est de s'assurer que les informations de cluster contenues dans le cache de référentiel complet existent à deux endroits à des fins de récupération. S'il n'existe qu'un seul référentiel complet et qu'il perd ses informations sur le cluster, une intervention manuelle sur tous les gestionnaires de files d'attente du cluster est nécessaire pour que le cluster fonctionne à nouveau. Toutefois, s'il existe deux référentiels complets, étant donné que les informations sont toujours publiées et souscrites à partir de deux référentiels complets, le référentiel complet ayant échoué peut être récupéré avec un minimum d'effort.

- Il est possible d'effectuer des opérations de maintenance sur les gestionnaires de files d'attente de référentiel complet dans une conception de cluster de référentiel complet sans affecter les utilisateurs de ce cluster: le cluster continue de fonctionner avec un seul référentiel, de sorte que, dans la mesure du possible, les référentiels sont arrêtés, la maintenance est appliquée et une nouvelle sauvegarde est effectuée à la fois. Même en cas d'indisponibilité sur le deuxième référentiel complet, les applications en cours d'exécution ne sont pas affectées pendant au moins trois jours.
- Sauf s'il existe une bonne raison d'utiliser un troisième référentiel, par exemple l'utilisation d'un référentiel complet géographiquement local pour des raisons géographiques, utilisez la conception de deux référentiels. Le fait de disposer de trois référentiels complets signifie que vous ne savez jamais quels sont les deux qui sont actuellement utilisés et qu'il peut y avoir des problèmes d'administration causés par des interactions entre plusieurs paramètres de gestion de charge de travail. Il n'est pas recommandé de disposer de plus de deux référentiels complets.
- Si vous avez encore besoin d'une meilleure disponibilité, pensez à héberger les gestionnaires de files d'attente de référentiel complet en tant que gestionnaires de files d'attente multi-instance ou à utiliser la prise en charge de la haute disponibilité spécifique à la plateforme pour améliorer leur disponibilité.
- Vous devez interconnecter complètement tous les gestionnaires de files d'attente de référentiel complet avec des canaux émetteurs de cluster définis manuellement. Une attention particulière doit être accordée lorsque le cluster possède, pour une raison légitime, plus de deux référentiels complets. Dans cette situation, il est souvent possible de passer à côté d'un ou de plusieurs canaux et de ne pas le voir immédiatement. Lorsqu'il n'y a pas d'interconnexion complète, il est souvent difficile de diagnostiquer les problèmes. Ils sont difficiles à diagnostiquer car certains référentiels complets ne contiennent pas toutes les données de référentiel et, par conséquent, les gestionnaires de files d'attente du cluster ont des vues différentes du cluster en fonction des référentiels complets auxquels ils se connectent.

## **Les applications devraient-elles utiliser des files d'attente dans les référentiels complets?**

Un référentiel complet est exactement comme n'importe quel autre gestionnaire de files d'attente et il est donc possible d'héberger des files d'attente d'application dans le référentiel complet et de connecter des applications directement à ces gestionnaires de files d'attente. Les applications devraient-elles utiliser des files d'attente dans les référentiels complets?

La réponse communément acceptée est "Non?". Bien que cette configuration soit possible, de nombreux clients préfèrent que ces gestionnaires de files d'attente soient dédiés à la gestion du cache de cluster de référentiel complet. Les points à prendre en compte pour décider de l'une ou l'autre des options sont décrits ici, mais en fin de compte, l'architecture de cluster doit être adaptée aux exigences particulières de l'environnement.

- Mises à niveau: En règle générale, afin d'utiliser les nouvelles fonctions de cluster dans les nouvelles éditions d' IBM MQ , les gestionnaires de files d'attente de référentiel complet de ce cluster doivent d'abord être mis à niveau. Lorsqu'une application du cluster souhaite utiliser de nouvelles fonctions, il peut être utile de pouvoir mettre à jour les référentiels complets (et certains sous-ensembles de référentiels partiels) sans tester un certain nombre d'applications colocalisés.
- Maintenance: De la même manière, si vous devez appliquer une maintenance urgente aux référentiels complets, ils peuvent être redémarrés ou actualisés à l'aide de la commande **REFRESH** sans toucher aux applications.
- Performances: au fur et à mesure que les clusters augmentent et que les demandes liées à la maintenance complète du cache de cluster de référentiel augmentent, le fait de conserver les applications séparément réduit le risque que cela affecte les performances de l'application par le biais de conflits de ressources système.
- Configuration matérielle requise: En règle générale, les référentiels complets n'ont pas besoin d'être puissants ; par exemple, un serveur UNIX simple avec une bonne attente de disponibilité est suffisant. Alternativement, pour les clusters de très grande taille ou en constante évolution, les performances de l'ordinateur de référentiel complet doivent être prises en compte.
- Configuration logicielle requise: les exigences sont généralement la raison principale pour choisir d'héberger des files d'attente d'application sur un référentiel complet. Dans un petit cluster, la colocalisation peut signifier que le nombre de gestionnaires de files d'attente/serveurs requis doit être réduit.

## Gestion des définitions de canal

Même au sein d'un même cluster, plusieurs définitions de canal peuvent exister en fournissant plusieurs routes entre deux gestionnaires de files d'attente.

Il y a parfois un avantage à avoir des canaux parallèles au sein d'un même cluster, mais cette décision de conception doit être examinée de manière approfondie ; en plus d'ajouter de la complexité, cette conception peut entraîner une sous-utilisation des canaux, ce qui réduit les performances. Cette situation est due au fait que les tests impliquent généralement l'envoi d'un grand nombre de messages à un débit constant, de sorte que les canaux parallèles sont pleinement utilisés. Mais avec les conditions réelles d'un flux de messages non constant, l'algorithme d'équilibrage de la charge de travail entraîne une baisse des performances à mesure que le flux de messages passe d'un canal à un autre.

Lorsqu'un gestionnaire de files d'attente est membre de plusieurs clusters, il est possible d'utiliser une définition de canal unique avec une liste de noms de cluster plutôt que de définir un canal CLUSRCVR distinct pour chaque cluster. Toutefois, cette configuration peut entraîner des difficultés d'administration ultérieurement ; prenez par exemple le cas où TLS doit être appliqué à un cluster mais pas à une seconde. Il est donc préférable de créer des définitions distinctes, et la convention de dénomination suggérée dans [«Conventions de dénomination de cluster»](#), à la page 36 le prend en charge.

## Équilibrage de la charge de travail sur plusieurs canaux

Ces informations sont destinées à une compréhension approfondie du sujet. Pour obtenir l'explication de base de ce sujet (qui doit être comprise avant d'utiliser les informations ici), voir [Utilisation de clusters pour la gestion de la charge de travail](#), [Équilibrage de la charge de travail dans les clusters](#) et [Algorithme de gestion de la charge de travail du cluster](#).

L'algorithme de gestion de la charge de travail du cluster fournit un large ensemble d'outils, mais ils ne doivent pas tous être utilisés les uns avec les autres sans comprendre parfaitement comment ils fonctionnent et interagissent. Il se peut que l'importance des canaux pour le processus d'équilibrage de la charge de travail ne soit pas immédiatement évidente: l'algorithme de permutation circulaire de la gestion de la charge de travail se comporte comme si plusieurs canaux de cluster d'un gestionnaire de files d'attente possédant une file d'attente en cluster étaient traités comme plusieurs instances de cette file d'attente. Ce processus est expliqué plus en détail dans l'exemple suivant:

1. Deux gestionnaires de files d'attente hébergent une file d'attente dans un cluster: QM1 et QM2.
2. Il existe cinq canaux récepteurs de cluster pour QM1.

3. Il n'existe qu'un seul canal récepteur de cluster pour QM2.
4. Lorsque **MQPUT** ou **MQOPEN** on QM3 choisit une instance, l'algorithme est cinq fois plus susceptible d'envoyer le message à QM1 qu'à QM2.
5. La situation à l'étape 4 se produit car l'algorithme voit six options à choisir (5 + 1) et des permutation circulaire sur les cinq canaux vers QM1 et le canal unique vers QM2.

Un autre comportement subtil est que, même lors de l'insertion de messages dans une file d'attente en cluster dont une instance est configurée sur le gestionnaire de files d'attente local, IBM MQ utilise l'état du canal récepteur de cluster local pour décider si les messages doivent être insérés dans l'instance locale de la file d'attente ou dans les instances distantes de la file d'attente. Dans ce scénario :

1. Lors de l'insertion de messages, l'algorithme de gestion de la charge de travail ne prend pas en compte les files d'attente de cluster individuelles, mais les canaux de cluster pouvant atteindre ces destinations.
2. Pour atteindre les destinations locales, les canaux récepteurs locaux sont inclus dans cette liste (bien qu'ils ne soient pas utilisés pour envoyer le message).
3. Lorsqu'un canal récepteur local est arrêté, l'algorithme de gestion de charge de travail préfère une autre instance par défaut si son CLUSRCVR n'est pas arrêté. S'il existe plusieurs instances CLUSRCVR locales pour la destination et qu'au moins une instance n'est pas arrêtée, l'instance locale reste éligible.

*Mise en cluster: isolement d'application à l'aide de plusieurs files d'attente de transmission de cluster*

Vous pouvez isoler les flux de messages entre les gestionnaires de files d'attente d'un cluster. Vous pouvez placer des messages transportés par des canaux émetteurs de cluster différents dans des files d'attente de transmission de cluster différentes. Vous pouvez utiliser l'approche dans un cluster unique ou avec des clusters qui se chevauchent. La rubrique fournit des exemples et des meilleures pratiques pour vous guider dans le choix d'une approche à utiliser.

Lorsque vous déployez une application, vous avez le choix entre les ressources IBM MQ qu'elle partage avec d'autres applications et celles qu'elle ne partage pas. Il existe un certain nombre de types de ressources pouvant être partagées, les principales étant le serveur lui-même, le gestionnaire de files d'attente, les canaux et les files d'attente. Vous pouvez choisir de configurer des applications avec moins de ressources partagées, en allouant des files d'attente, des canaux, des gestionnaires de files d'attente ou même des serveurs distincts à des applications individuelles. Dans ce cas, la configuration globale du système devient plus grande et plus complexe. L'utilisation de clusters IBM MQ réduit la complexité de la gestion d'un plus grand nombre de serveurs, de gestionnaires de files d'attente, de files d'attente et de canaux, mais elle introduit une autre ressource partagée, la file d'attente de transmission de cluster, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

Figure 12, à la page 53 est une tranche d'un déploiement IBM MQ de grande taille qui illustre la signification du partage `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Dans le diagramme, l'application, `Client App`, est connectée au gestionnaire de files d'attente QM2 dans le cluster CL1. Un message provenant de `Client App` est traité par l'application, `Server App`. Le message est extrait par `Server App` de la file d'attente de cluster Q1 sur le gestionnaire de files d'attente QM3 dans CLUSTER2. Etant donné que les applications client et serveur ne se trouvent pas dans le même cluster, le message est transféré par le gestionnaire de files d'attente de passerelle QM1.

La méthode normale pour configurer une passerelle de cluster consiste à faire du gestionnaire de files d'attente de passerelle un membre de tous les clusters. Sur le gestionnaire de files d'attente de passerelle, des files d'attente d'alias de cluster sont définies pour les files d'attente de cluster dans tous les clusters. Les alias de file d'attente en cluster sont disponibles dans tous les clusters. Les messages insérés dans les alias de file d'attente de cluster sont acheminés via le gestionnaire de files d'attente de passerelle vers leur destination correcte. Le gestionnaire de files d'attente de passerelle place les messages envoyés aux files d'attente d'alias de cluster dans le `SYSTEM.CLUSTER.TRANSMIT.QUEUE` commun sur QM1.

L'architecture concentrateur et satellite requiert que tous les messages entre les clusters transitent par le gestionnaire de files d'attente de passerelle. Il en résulte que tous les messages transitent par la file d'attente de transmission de cluster unique sous QM1, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.

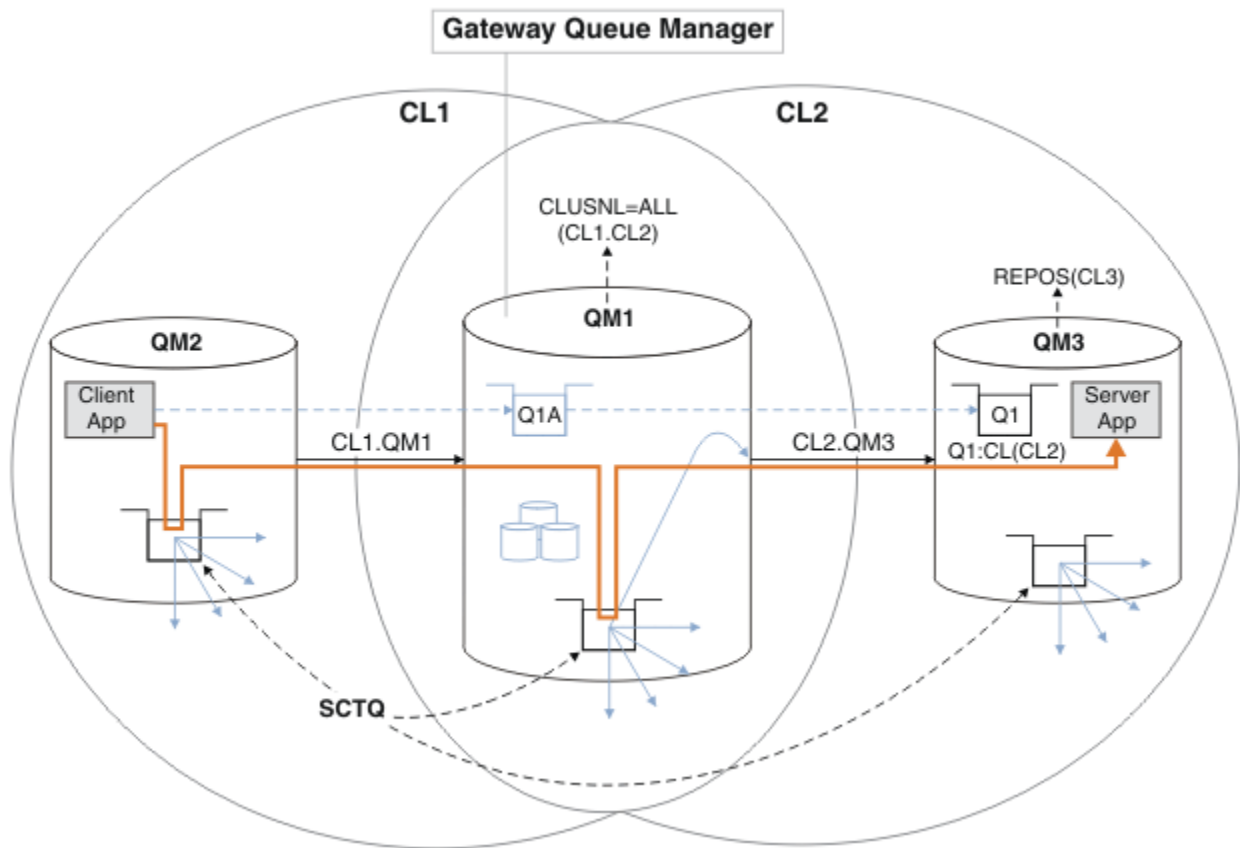
Du point de vue des performances, une file d'attente unique n'est pas un problème. Une file d'attente de transmission commune ne représente généralement pas un goulot d'étranglement des performances. Le débit des messages sur la passerelle est largement déterminé par les performances des canaux qui s'y connectent. Le débit n'est généralement pas affecté par le nombre de files d'attente ou le nombre de messages dans les files d'attente qui utilisent les canaux.

D'autres points de vue, l'utilisation d'une file d'attente de transmission unique pour plusieurs applications présente les inconvénients suivants:

- Vous ne pouvez pas isoler le flux de messages vers une destination à partir du flux de messages vers une autre destination. Vous ne pouvez pas séparer le stockage des messages avant qu'ils ne soient réacheminés, même si les destinations se trouvent dans des clusters différents sur des gestionnaires de files d'attente différents.

Si une destination de cluster devient indisponible, les messages de cette destination sont générés dans la file d'attente de transmission unique, puis les messages sont remplis. Une fois que la file d'attente de transmission est saturée, elle empêche les messages d'être placés dans la file d'attente de transmission pour n'importe quelle destination de cluster.

- Il n'est pas facile de surveiller le transfert de messages vers différentes destinations de cluster. Tous les messages se trouvent dans la file d'attente de transmission unique. L'affichage de la longueur de la file d'attente de transmission vous donne peu d'indication sur le transfert des messages vers toutes les destinations.



**Remarque :** Les flèches dans [Figure 12](#), à la [page 53](#) et les figures suivantes sont de types différents. Les flèches pleines représentent les flux de messages. Les libellés sur les flèches pleines sont des noms de canal de message. Les flèches pleines grises sont des flux de messages potentiels du SYSTEM . CLUSTER . TRANSMIT . QUEUE vers les canaux émetteurs de cluster. Des lignes en pointillés noirs relient les étiquettes à leurs cibles. Les flèches en pointillés grises sont des références ; par exemple, à partir d'un appel MQOPEN par Client App à la définition de file d'attente d'alias de cluster Q1A.

*Figure 12. Application client-serveur déployée sur l'architecture concentrateur et satellite à l'aide de clusters IBM MQ*

Dans [Figure 12](#), à la [page 53](#), les clients de Server App ouvrent la file d'attente Q1A. Les messages sont placés dans SYSTEM . CLUSTER . TRANSMIT . QUEUE sur QM2, transférés dans SYSTEM . CLUSTER . TRANSMIT . QUEUE sur QM1, puis transférés dans Q1 sur QM3, où ils sont reçus par l'application Server App .

Le message de Client App passe par les files d'attente de transmission de cluster système sous QM2 et QM1. Dans [Figure 12](#), à la [page 53](#), l'objectif est d'isoler le flux de messages sur le gestionnaire de files d'attente de passerelle de l'application client afin que ses messages ne soient pas stockés sur SYSTEM . CLUSTER . TRANSMIT . QUEUE. Vous pouvez isoler les flux sur n'importe lequel des autres gestionnaires de files d'attente en cluster. Vous pouvez également isoler les flux dans l'autre sens, vers le client. Pour que les descriptions des solutions soient brèves, elles ne prennent en compte qu'un seul flux provenant de l'application client.

### **Solutions permettant d'isoler le trafic des messages de cluster sur un gestionnaire de files d'attente de passerelle de cluster**

Une façon de résoudre le problème consiste à utiliser des alias de gestionnaire de files d'attente, ou des définitions de files d'attente éloignées, pour établir une passerelle entre les clusters. Créez une définition de file d'attente éloignée en cluster, une file d'attente de transmission et un canal pour séparer chaque flux de messages sur le gestionnaire de files d'attente de passerelle. Voir [Ajout d'une définition](#)

de file d'attente éloignée pour isoler les messages envoyés à partir d'un gestionnaire de files d'attente de passerelle.

A partir de la IBM WebSphere MQ 7.5 , les gestionnaires de files d'attente de cluster ne sont pas limités à une seule file d'attente de transmission de cluster. Vous avez deux possibilités :

1. Définissez manuellement des files d'attente de transmission de cluster supplémentaires et définissez les canaux émetteurs de cluster qui transfèrent les messages de chaque file d'attente de transmission. Voir Ajout d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle.
2. Permet au gestionnaire de files d'attente de créer et de gérer automatiquement des files d'attente de transmission de cluster supplémentaires. Il définit une file d'attente de transmission de cluster différente pour chaque canal émetteur de cluster ; voir Modification de la valeur par défaut pour séparer les files d'attente de transmission de cluster afin d'isoler le trafic des messages.

Vous pouvez combiner des files d'attente de transmission de cluster définies manuellement pour certains canaux émetteurs de cluster avec le gestionnaire de files d'attente qui gère le reste. La combinaison des files d'attente de transmission est l'approche utilisée dans la rubrique Ajout d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle. Dans cette solution, la plupart des messages entre les clusters utilisent le SYSTEM . CLUSTER . TRANSMIT . QUEUEcommun. Une application est critique et tous ses flux de messages sont isolés des autres flux à l'aide d'une file d'attente de transmission de cluster définie manuellement.

La configuration de la rubrique Ajout d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle est limitée. Il ne sépare pas le trafic de messages vers une file d'attente de cluster sur le même gestionnaire de files d'attente dans le même cluster qu'une autre file d'attente de cluster. Vous pouvez séparer le trafic de messages vers des files d'attente individuelles à l'aide des définitions de file d'attente éloignée qui font partie de la mise en file d'attente répartie. Avec les clusters, qui utilisent plusieurs files d'attente de transmission de cluster, vous pouvez séparer le trafic de messages qui est acheminé vers différents canaux émetteurs de cluster. Plusieurs files d'attente de cluster dans le même cluster, sur le même gestionnaire de files d'attente, partagent un canal émetteur de cluster. Les messages de ces files d'attente sont stockés dans la même file d'attente de transmission, avant d'être réacheminés à partir du gestionnaire de files d'attente de passerelle. Dans la configuration de la rubrique Ajout d'un cluster et d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle, la limitation consiste à ajouter un autre cluster et à faire du gestionnaire de files d'attente et de la file d'attente de cluster un membre du nouveau cluster. Le nouveau gestionnaire de files d'attente peut être le seul gestionnaire de files d'attente du cluster. Vous pouvez ajouter d'autres gestionnaires de files d'attente au cluster et utiliser le même cluster pour isoler également les files d'attente de cluster sur ces gestionnaires de files d'attente.

### **Concepts associés**

«Contrôle d'accès et files d'attente de transmission de cluster multiples», à la page 30

Choisissez entre trois modes de vérification lorsqu'une application place des messages dans des files d'attente de cluster éloignées. Les modes sont la vérification à distance par rapport à la file d'attente de cluster, la vérification en local par rapport à SYSTEM . CLUSTER . TRANSMIT . QUEUEou la vérification par rapport aux profils locaux pour la file d'attente de cluster ou le gestionnaire de files d'attente de cluster.

Utilisation de files d'attente de transmission de cluster et de canaux émetteurs de cluster

«Chevauchement de clusters», à la page 38

Les clusters qui se chevauchent fournissent des fonctions d'administration supplémentaires. Utilisez des listes de noms pour réduire le nombre de commandes nécessaires à l'administration des clusters qui se chevauchent.

### **Tâches associées**

Autorisation d'insertion de messages dans des files d'attente de cluster éloignées

Ajout d'une définition de file d'attente éloignée pour isoler les messages envoyés à partir d'un gestionnaire de files d'attente de passerelle

Ajout d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle

[Ajout d'un cluster et d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle](#)

[Modification de la valeur par défaut pour séparer les files d'attente de transmission de cluster afin d'isoler le trafic des messages](#)

[Création de clusters à double chevauchement avec un gestionnaire de files d'attente de passerelle](#)

[Configuration des chemins d'accès aux messages entre les clusters](#)

[Sécurisation](#)

### Référence associée

[setmqaut](#)

*Mise en cluster: Planification de la configuration des files d'attente de transmission de cluster*

Vous êtes guidé dans les choix de files d'attente de transmission de cluster. Vous pouvez configurer une file d'attente par défaut commune, des files d'attente par défaut distinctes ou des files d'attente définies manuellement.

## Avant de commencer

Consultez [«Comment choisir le type de file d'attente de transmission de cluster à utiliser»](#), à la page 58.

## Pourquoi et quand exécuter cette tâche

Lorsque vous planifiez la configuration d'un gestionnaire de files d'attente pour sélectionner une file d'attente de transmission de cluster, vous avez le choix entre plusieurs options.

1. Quelle est la file d'attente de transmission de cluster par défaut pour les transferts de messages de cluster?
  - a. Une file d'attente de transmission de cluster commune, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`.
  - b. Des files d'attente de transmission de cluster distinctes. Le gestionnaire de files d'attente gère les files d'attente de transmission de cluster distinctes. Il les crée en tant que files d'attente dynamiques permanentes à partir de la file d'attente modèle, `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE`. Il crée une file d'attente de transmission de cluster pour chaque canal émetteur de cluster qu'il utilise.
2. Pour les files d'attente de transmission de cluster que vous décidez de créer manuellement, vous disposez de deux options supplémentaires:
  - a. Définissez une file d'attente de transmission distincte pour chaque canal émetteur de cluster que vous décidez de configurer manuellement. Dans ce cas, définissez l'attribut de file d'attente **CLCHNAME** de la file d'attente de transmission sur le nom d'un canal émetteur de cluster. Sélectionnez le canal émetteur de cluster qui doit transférer les messages à partir de cette file d'attente de transmission.
  - b. Combinez le trafic de messages pour un groupe de canaux émetteurs de cluster dans la même file d'attente de transmission de cluster ; voir [Figure 13](#), à la page 56. Dans ce cas, définissez l'attribut de file d'attente **CLCHNAME** de chaque file d'attente de transmission commune sur un nom de canal émetteur de cluster générique. Un nom de canal émetteur de cluster générique est un filtre permettant de regrouper les noms de canal émetteur de cluster. Par exemple, `SALES.*` regroupe tous les canaux émetteurs de cluster dont le nom commence par `SALES.` . Vous pouvez placer plusieurs caractères génériques n'importe où dans la chaîne de filtrage. Le caractère générique est un astérisque, `*`. Il représente de zéro à n'importe quel nombre de caractères.

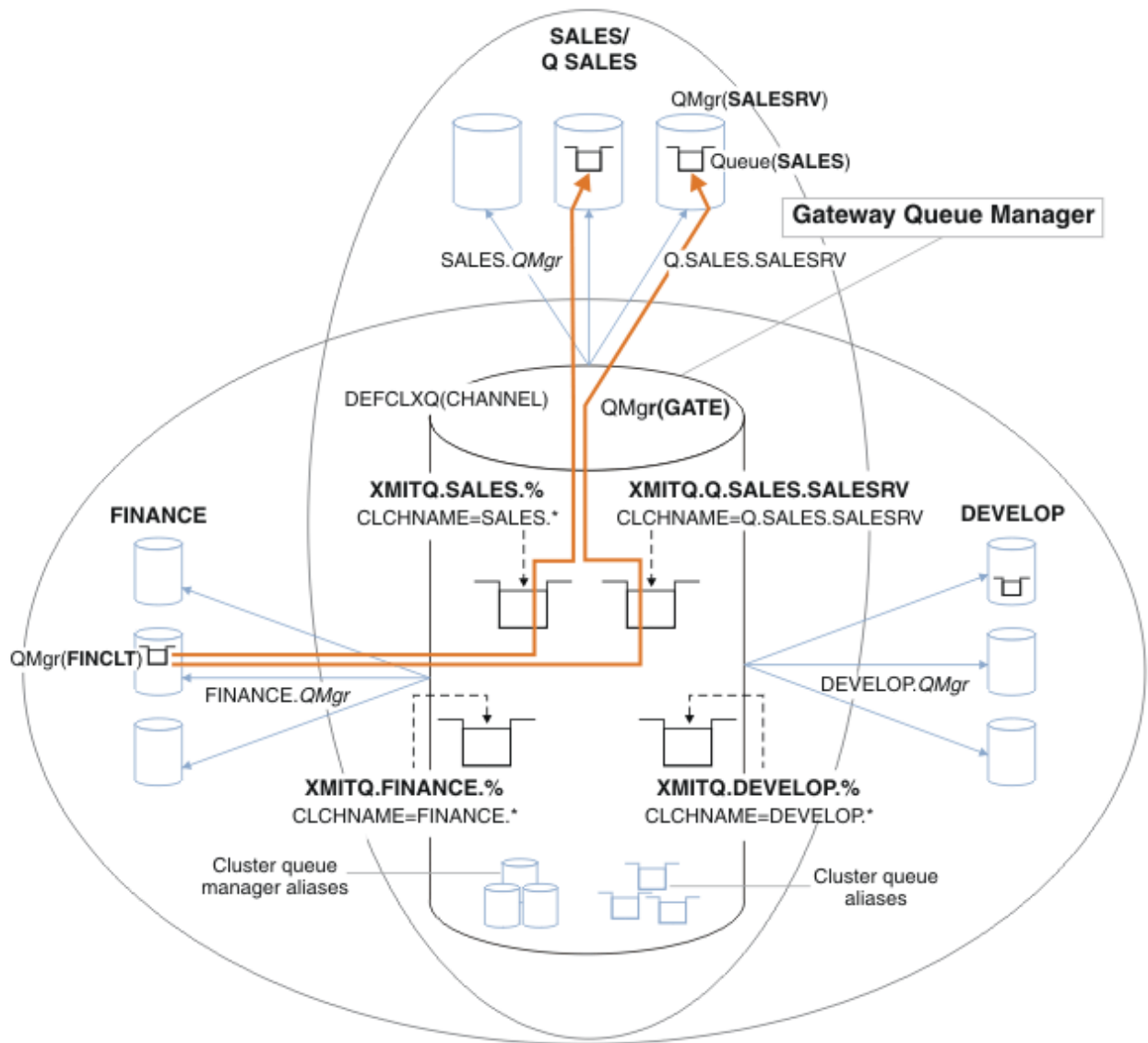


Figure 13. Exemple de files d'attente de transmission spécifiques pour différents clusters IBM MQ de service

## Procédure

1. Sélectionnez le type de file d'attente de transmission de cluster par défaut à utiliser.

- Choisissez une file d'attente de transmission de cluster unique ou des files d'attente distinctes pour chaque connexion de cluster.

Conservez le paramètre par défaut ou exécutez la commande **MQSC** :

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

2. Isolez les flux de messages qui ne doivent pas partager une file d'attente de transmission de cluster avec d'autres flux.

- Voir «Mise en cluster: exemple de configuration de plusieurs files d'attente de transmission de cluster», à la page 60. Dans l'exemple, la file d'attente SALES, qui doit être isolée, est un membre du cluster SALES, sur SALESRV. Pour isoler la file d'attente SALES, créez un cluster Q.SALES, faites du gestionnaire de files d'attente SALESRV un membre et modifiez la file d'attente SALES pour qu'elle appartienne à Q.SALES.



- Les gestionnaires de files d'attente qui envoient des messages à SALES doivent également être membres du nouveau cluster. Si vous utilisez un alias de file d'attente en cluster et un gestionnaire de files d'attente de passerelle, comme dans l'exemple, dans de nombreux cas, vous pouvez limiter les modifications pour faire du gestionnaire de files d'attente de passerelle un membre du nouveau cluster.
- Toutefois, la séparation des flux de la passerelle vers la destination ne sépare pas les flux de la passerelle du gestionnaire de files d'attente source. Mais il s'avère parfois suffisant pour séparer les flux de la passerelle et non les flux vers la passerelle. Si ce n'est pas suffisant, ajoutez le gestionnaire de files d'attente source dans le nouveau cluster. Si vous souhaitez que les messages transitent par la passerelle, déplacez l'alias de cluster vers le nouveau cluster et continuez à envoyer des messages à l'alias de cluster sur la passerelle, et non pas directement vers le gestionnaire de files d'attente cible.

Pour isoler les flux de messages, procédez comme suit:

- a) Configurez les destinations des flux de sorte que chaque file d'attente cible soit la seule file d'attente d'un cluster spécifique, sur ce gestionnaire de files d'attente.
  - b) Créez les canaux émetteur et récepteur de cluster pour tous les nouveaux clusters que vous avez créés en suivant une convention de dénomination systématique.
    - Voir «[Mise en cluster: considérations spéciales pour le chevauchement de clusters](#)», à la page 46.
  - c) Définissez une file d'attente de transmission de cluster pour chaque destination isolée sur chaque gestionnaire de files d'attente qui envoie des messages à la file d'attente cible.
    - Une convention de dénomination pour les files d'attente de transmission de cluster consiste à utiliser la valeur de l'attribut de nom de canal de cluster, CLCHNAME, avec le préfixe XMITQ.
3. Créez des files d'attente de transmission de cluster pour répondre aux exigences de gouvernance ou de surveillance.
- Les exigences de gouvernance et de surveillance standard génèrent une file d'attente de transmission par cluster ou une file d'attente de transmission par gestionnaire de files d'attente. Si vous respectez la convention de dénomination des canaux de cluster, *ClusterName* . *QueueManagerName*, il est facile de créer des noms de canal génériques qui sélectionnent un cluster de gestionnaires de files d'attente ou tous les clusters dont un gestionnaire de files d'attente est membre ; voir «[Mise en cluster: exemple de configuration de plusieurs files d'attente de transmission de cluster](#)», à la page 60.
  - Étendez la convention de dénomination pour les files d'attente de transmission de cluster afin de prendre en charge les noms de canal génériques, en remplaçant le symbole de l'astérisque par un signe de pourcentage. Exemple :

```
DEFINE QLOCAL(XMITQ.SALES.%) USAGE(XMITQ) CLCHNAME(SALES.*)
```

## Concepts associés

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

«[Contrôle d'accès et files d'attente de transmission de cluster multiples](#)», à la page 30

Choisissez entre trois modes de vérification lorsqu'une application place des messages dans des files d'attente de cluster éloignées. Les modes sont la vérification à distance par rapport à la file d'attente de cluster, la vérification en local par rapport à SYSTEM.CLUSTER.TRANSMIT.QUEUE ou la vérification par rapport aux profils locaux pour la file d'attente de cluster ou le gestionnaire de files d'attente de cluster.

«[Chevauchement de clusters](#)», à la page 38

Les clusters qui se chevauchent fournissent des fonctions d'administration supplémentaires. Utilisez des listes de noms pour réduire le nombre de commandes nécessaires à l'administration des clusters qui se chevauchent.

## Tâches associées

[Ajout d'une définition de file d'attente éloignée pour isoler les messages envoyés à partir d'un gestionnaire de files d'attente de passerelle](#)

Ajout d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle

Ajout d'un cluster et d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle

Modification de la valeur par défaut pour séparer les files d'attente de transmission de cluster afin d'isoler le trafic des messages

Création de clusters à double chevauchement avec un gestionnaire de files d'attente de passerelle

Configuration des chemins d'accès aux messages entre les clusters

*Comment choisir le type de file d'attente de transmission de cluster à utiliser*

Comment choisir entre différentes options de configuration de file d'attente de transmission de cluster.

Vous pouvez choisir quelle file d'attente de transmission de cluster est associée à un canal émetteur de cluster.

1. Tous les canaux émetteurs de cluster peuvent être associés à la file d'attente de transmission de cluster par défaut unique, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`; cette option est la valeur par défaut.
2. Vous pouvez définir tous les canaux émetteurs de cluster pour qu'ils soient automatiquement associés à une file d'attente de transmission de cluster distincte. Les files d'attente sont créées par le gestionnaire de files d'attente à partir de la file d'attente modèle `SYSTEM.CLUSTER.TRANSMIT.MODEL.QUEUE` et nommées `SYSTEM.CLUSTER.TRANSMIT.ChannelName`. Les canaux utilisent leur file d'attente de transmission de cluster à nom unique si l'attribut de gestionnaire de files d'attente **DEFCLXQ** est défini sur `CHANNEL`.
3. Vous pouvez définir des canaux émetteurs de cluster spécifiques pour qu'ils soient servis par une file d'attente de transmission de cluster unique. Sélectionnez cette option en créant une file d'attente de transmission et en définissant son attribut **CLCHNAME** sur le nom du canal émetteur de cluster.
4. Vous pouvez sélectionner des groupes de canaux émetteurs de cluster à servir par une file d'attente de transmission de cluster unique. Sélectionnez cette option en créant une file d'attente de transmission et en définissant son attribut **CLCHNAME** sur un nom de canal générique, tel que `ClusterName.*`. Si vous nommez des canaux de cluster en suivant les conventions de dénomination de la rubrique «Mise en cluster: considérations spéciales pour le chevauchement de clusters», à la page 46, ce nom sélectionne tous les canaux de cluster connectés aux gestionnaires de files d'attente du cluster `ClusterName`.

Vous pouvez combiner l'une des options de file d'attente de transmission de cluster par défaut pour certains canaux émetteurs de cluster avec un certain nombre de configurations de file d'attente de transmission de cluster spécifiques et génériques.

## Valeurs recommandées

Dans la plupart des cas, pour les installations IBM MQ existantes, la configuration par défaut est le meilleur choix. Un gestionnaire de files d'attente de cluster stocke les messages de cluster dans une file d'attente de transmission de cluster unique, `SYSTEM.CLUSTER.TRANSMIT.QUEUE`. Vous avez la possibilité de modifier la valeur par défaut pour stocker des messages pour différents gestionnaires de files d'attente et différents clusters dans des files d'attente de transmission distinctes, ou de définir vos propres files d'attente de transmission.

Dans la plupart des cas, pour les nouvelles installations IBM MQ, la configuration par défaut est également le meilleur choix. Le processus de passage de la configuration par défaut à la valeur par défaut alternative, qui consiste à disposer d'une file d'attente de transmission pour chaque canal émetteur de cluster, est automatique. Le retour en arrière est également automatique. Le choix de l'un ou de l'autre n'est pas critique, vous pouvez l'inverser.

Le choix d'une configuration différente est davantage lié à la gouvernance et à la gestion qu'à la fonctionnalité ou aux performances. A quelques exceptions près, la configuration de plusieurs files d'attente de transmission de cluster ne profite pas au comportement du gestionnaire de files d'attente. Elle génère davantage de files d'attente et vous oblige à modifier les procédures de surveillance et de gestion que vous avez déjà configurées et qui font référence à la file d'attente de transmission unique.

C'est pourquoi, dans l'ensemble, le maintien de la configuration par défaut est le meilleur choix, sauf si vous avez des raisons de gouvernance ou de gestion fortes pour un choix différent.

Les deux exceptions concernent ce qui se passe si le nombre de messages stockés sur `SYSTEM . CLUSTER . TRANSMIT . QUEUE` augmente. Si vous effectuez chaque étape pour séparer les messages d'une destination des messages d'une autre destination, les problèmes de canal et de distribution avec une destination ne doivent pas affecter la distribution à une autre destination. Toutefois, le nombre de messages stockés sur `SYSTEM . CLUSTER . TRANSMIT . QUEUE` peut augmenter en raison de l'absence de distribution de messages suffisamment rapide à une destination. Le nombre de messages sur `SYSTEM . CLUSTER . TRANSMIT . QUEUE` pour une destination peut affecter la distribution des messages à d'autres destinations.

Pour éviter les problèmes résultant du remplissage d'une seule file d'attente de transmission, vous devez créer une capacité suffisante dans votre configuration. Ensuite, si une destination échoue et qu'un journal des messages en attente commence à s'accumuler, vous avez le temps de résoudre le problème.

Si les messages sont acheminés via un gestionnaire de files d'attente concentrateur, tel qu'une passerelle de cluster, ils partagent une file d'attente de transmission commune, `SYSTEM . CLUSTER . TRANSMIT . QUEUE`. Si le nombre de messages stockés dans `SYSTEM . CLUSTER . TRANSMIT . QUEUE` sur le gestionnaire de files d'attente de passerelle atteint sa longueur maximale, le gestionnaire de files d'attente commence à rejeter les nouveaux messages pour la file d'attente de transmission jusqu'à ce que cette longueur soit réduite. La congestion affecte les messages pour toutes les destinations qui sont routées via la passerelle. Les messages sauvegardés les files d'attente de transmission des autres gestionnaires de files d'attente qui envoient des messages à la passerelle. Le problème se manifeste dans les messages écrits dans les journaux d'erreurs du gestionnaire de files d'attente, avec une baisse du débit des messages et des temps écoulés plus longs entre l'envoi d'un message et l'arrivée d'un message à sa destination.

L'effet de la congestion sur une seule file d'attente de transmission peut se manifester, avant même qu'elle ne soit pleine. Si vous avez un trafic de messages mixte, avec des messages non persistants de grande taille et des messages de petite taille, le temps de distribution des messages de petite taille augmente à mesure que la file d'attente de transmission se remplit. Ce délai est dû à l'écriture sur le disque de messages non persistants volumineux qui ne seraient normalement pas écrits sur le disque. Si vous disposez de flux de messages critiques, partageant une file d'attente de transmission de cluster avec d'autres flux de messages mixtes, il peut être utile de configurer un chemin de messages spécial pour isoler des autres flux de messages ; voir [Ajout d'un cluster et d'une file d'attente de transmission de cluster pour isoler le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle](#).

Les autres raisons de la configuration de files d'attente de transmission de cluster distinctes sont le respect des exigences de gouvernance ou la simplification des messages de surveillance envoyés à des destinations de cluster différentes. Par exemple, vous pouvez être amené à démontrer que les messages d'une destination ne partagent jamais une file d'attente de transmission avec les messages d'une autre destination.

Modifiez l'attribut de gestionnaire de files d'attente **DEFCLXQ** qui contrôle la file d'attente de transmission de cluster par défaut afin de créer des files d'attente de transmission de cluster différentes pour chaque canal émetteur de cluster. Plusieurs destinations peuvent partager un canal émetteur de cluster, vous devez donc planifier vos clusters pour atteindre pleinement cet objectif. Appliquez la méthode [Ajout d'un cluster et d'une file d'attente de transmission de cluster pour isoler systématiquement le trafic de messages de cluster envoyé à partir d'un gestionnaire de files d'attente de passerelle](#) à toutes les files d'attente de votre cluster. Le résultat que vous visez est qu'aucune destination de cluster ne partage un canal émetteur de cluster avec une autre destination de cluster. Par conséquent, aucun message pour une destination de cluster ne partage sa file d'attente de transmission de cluster avec un message pour une autre destination.

La création d'une file d'attente de transmission de cluster distincte pour un flux de messages spécifique facilite la surveillance du flux de messages vers cette destination. Pour utiliser une nouvelle file d'attente de transmission de cluster, définissez la file d'attente, associez-la à un canal émetteur de cluster, puis arrêtez et démarrez le canal. Il n'est pas nécessaire que le changement soit permanent. Vous pouvez

isoler un flux de messages pendant un certain temps afin de surveiller la file d'attente de transmission, puis revenir à l'utilisation de la file d'attente de transmission par défaut.

### **Tâches associées**

Mise en cluster: exemple de configuration de plusieurs files d'attente de transmission de cluster

Dans cette tâche, vous appliquez les étapes de planification de plusieurs files d'attente de transmission de cluster à trois clusters qui se chevauchent. Les exigences sont de séparer les flux de messages vers une file d'attente de cluster, à partir de tous les autres flux de messages, et de stocker les messages de différents clusters dans différentes files d'attente de transmission de cluster.

Mise en cluster: changement des files d'attente de transmission de cluster

Planifiez la manière dont les modifications apportées aux files d'attente de transmission de cluster d'un gestionnaire de files d'attente de production existant seront prises en compte.

*Mise en cluster: exemple de configuration de plusieurs files d'attente de transmission de cluster*

Dans cette tâche, vous appliquez les étapes de planification de plusieurs files d'attente de transmission de cluster à trois clusters qui se chevauchent. Les exigences sont de séparer les flux de messages vers une file d'attente de cluster, à partir de tous les autres flux de messages, et de stocker les messages de différents clusters dans différentes files d'attente de transmission de cluster.

### **Pourquoi et quand exécuter cette tâche**

Les étapes de cette tâche montrent comment appliquer la procédure dans «Mise en cluster: Planification de la configuration des files d'attente de transmission de cluster», à la page 55 et arriver à la configuration illustrée dans Figure 14, à la page 61. Il s'agit d'un exemple de trois clusters qui se chevauchent, avec un gestionnaire de files d'attente de passerelle, configuré avec des files d'attente de transmission de cluster distinctes. Les commandes MQSC permettant de définir les clusters sont décrites dans «Création des exemples de clusters», à la page 63.

Pour l'exemple, il existe deux exigences. L'une consiste à séparer le flux de messages du gestionnaire de files d'attente de la passerelle vers l'application de vente qui consigne les ventes. La seconde consiste à demander combien de messages sont en attente d'être envoyés à différents secteurs ministériels à un moment donné. Les clusters SALES, FINANCE et DEVELOP sont déjà définis. Les messages de cluster sont actuellement transférés depuis SYSTEM.CLUSTER.TRANSMIT.QUEUE.

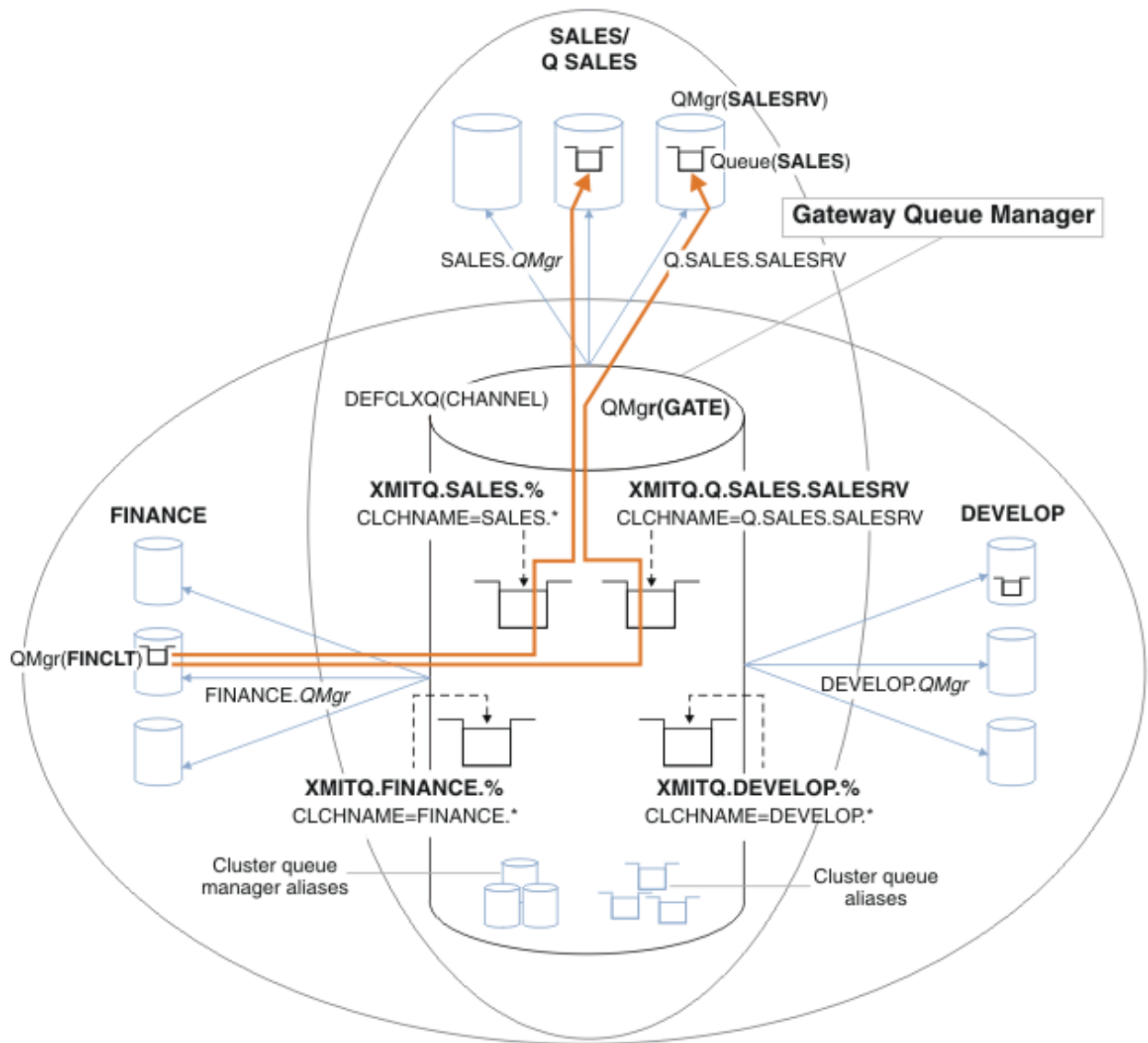


Figure 14. Exemple de files d'attente de transmission spécifiques pour différents clusters IBM MQ de service

Les étapes de modification des clusters sont les suivantes. Pour les définitions, voir [Modifications apportées pour isoler la file d'attente des ventes dans un nouveau cluster et séparer les files d'attente de transmission du cluster de passerelle](#).

## Procédure

1. La première étape de configuration consiste à " [Sélectionnez le type de file d'attente de transmission de cluster par défaut à utiliser](#) ".

La décision consiste à créer des files d'attente de transmission de cluster par défaut distinctes en exécutant la commande **MQSC** suivante sur le gestionnaire de files d'attente GATE .

```
ALTER QMGR DEFCLXQ(CHANNEL)
```

Il n'y a aucune raison de choisir cette valeur par défaut, car l'intention est de définir manuellement des files d'attente de transmission de cluster. Le choix a une valeur de diagnostic faible. Si une définition manuelle est effectuée de manière incorrecte et qu'un message circule dans une file d'attente de transmission de cluster par défaut, elle s'affiche lors de la création d'une file d'attente de transmission de cluster dynamique permanente.

2. La deuxième étape de configuration consiste à "Isolez les flux de messages qui ne doivent pas partager une file d'attente de transmission de cluster avec d'autres flux".

Dans ce cas, l'application de vente qui reçoit les messages de la file d'attente SALES sur SALESRV doit être isolée. Seul l'isolement des messages du gestionnaire de files d'attente de passerelle est requis. Les trois sous-étapes sont les suivantes:

- a) "Configurez les destinations des flux de sorte que chaque file d'attente cible soit la seule file d'attente d'un cluster spécifique, sur ce gestionnaire de files d'attente".

Cet exemple nécessite l'ajout du gestionnaire de files d'attente SALESRV à un nouveau cluster au sein du service des ventes. Si vous disposez de quelques files d'attente nécessitant un isolement, vous pouvez décider de créer un cluster spécifique pour la file d'attente SALES. Une convention de dénomination possible pour le nom de cluster consiste à nommer de tels clusters, Q. *QueueName*, par exemple Q.SALES. Une autre approche, qui peut être plus pratique si vous avez un grand nombre de files d'attente à isoler, consiste à créer des clusters de files d'attente isolées où et quand cela est nécessaire. Les noms des clusters peuvent être QUEUES. *n*.

Dans l'exemple, le nouveau cluster est appelé Q.SALES. Pour ajouter le nouveau cluster, voir les définitions dans Modifications permettant d'isoler la file d'attente des ventes dans un nouveau cluster et de séparer les files d'attente de transmission du cluster de passerelle. Le récapitulatif des modifications apportées à la définition est le suivant:

- i) Ajoutez Q.SALES à la liste de noms des clusters sur les gestionnaires de files d'attente de référentiel. La liste de noms est référencée dans le paramètre **REPOSNL** du gestionnaire de files d'attente.
- ii) Ajoutez Q.SALES à la liste de noms des clusters sur le gestionnaire de files d'attente de passerelle. La liste de noms est référencée dans toutes les définitions d'alias de file d'attente de cluster et d'alias de gestionnaire de files d'attente de cluster sur le gestionnaire de files d'attente de passerelle.
- iii) Créez une liste de noms sur le gestionnaire de files d'attente SALESRV, pour les deux clusters dont il est membre, et modifiez l'appartenance à un cluster de la file d'attente SALES :

```
DEFINE NAMLIST(CLUSTERS) NAMES(SALES, Q.SALES) REPLACE  
ALTER QLOCAL(SALES) CLUSTER(' ') CLUSNL(SALESRV.CLUSTERS)
```

La file d'attente SALES est membre des deux clusters, uniquement pour la transition. Une fois la nouvelle configuration en cours d'exécution, vous supprimez la file d'attente SALES du cluster SALES ; voir [Figure 15](#), à la page 66.

- b) "Créez les canaux émetteur et récepteur de cluster pour tous les nouveaux clusters que vous avez créés en suivant une convention de dénomination systématique".
- i) Ajoutez le canal récepteur de cluster Q.SALES. *RepositoryQMGr* à chacun des gestionnaires de files d'attente de référentiel
  - ii) Ajoutez le canal émetteur de cluster Q.SALES. *OtherRepositoryQMGr* à chacun des gestionnaires de files d'attente de référentiel pour vous connecter à l'autre gestionnaire de référentiels. Démarrez ces canaux.
  - iii) Ajoutez les canaux récepteurs de cluster Q.SALES.SALESRVet Q.SALES.GATE à l'un des gestionnaires de files d'attente de référentiel en cours d'exécution.
  - iv) Ajoutez les canaux émetteurs de cluster Q.SALES.SALESRVet Q.SALES.GATE aux gestionnaires de files d'attente SALESRV et GATE. Connectez le canal émetteur de cluster au gestionnaire de files d'attente de référentiel sur lequel vous avez créé les canaux récepteurs de cluster.
- c) "Définissez une file d'attente de transmission de cluster pour chaque destination isolée sur chaque gestionnaire de files d'attente qui envoie des messages à la file d'attente cible".

Sur le gestionnaire de files d'attente de passerelle, définissez la file d'attente de transmission de cluster XMITQ.Q.SALES.SALESRV pour le canal émetteur de cluster Q.SALES.SALESRV :

```
DEFINE QLOCAL(XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
```

3. La troisième étape de configuration consiste à "Créez des files d'attente de transmission de cluster pour répondre aux exigences de gouvernance ou de surveillance".

Sur le gestionnaire de files d'attente de passerelle, définissez les files d'attente de transmission de cluster:

```
DEFINE QLOCAL(XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE  
DEFINE QLOCAL(XMITQ.DEVELOP) USAGE(XMITQ) CLCHNAME(DEVELOP.*) REPLACE  
DEFINE QLOCAL(XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
```

## Que faire ensuite

Basculez vers la nouvelle configuration sur le gestionnaire de files d'attente de passerelle.

Le commutateur est déclenché en démarrant les nouveaux canaux et en redémarrant les canaux qui sont maintenant associés à des files d'attente de transmission différentes. Vous pouvez également arrêter et démarrer le gestionnaire de files d'attente de passerelle.

1. Arrêtez les canaux suivants sur le gestionnaire de files d'attente de passerelle:

```
SALES. Qmgr  
DEVELOP. Qmgr  
FINANCE. Qmgr
```

2. Démarrez les canaux suivants sur le gestionnaire de files d'attente de passerelle:

```
SALES. Qmgr  
DEVELOP. Qmgr  
FINANCE. Qmgr  
Q.SALES.SAVESRV
```

Une fois le commutateur terminé, supprimez la file d'attente SALES du cluster SALES ; voir [Figure 15](#), à la [page 66](#).

## Concepts associés

[Comment choisir le type de file d'attente de transmission de cluster à utiliser](#)

[Comment choisir entre différentes options de configuration de file d'attente de transmission de cluster.](#)

## Tâches associées

[Mise en cluster: changement des files d'attente de transmission de cluster](#)

[Planifiez la manière dont les modifications apportées aux files d'attente de transmission de cluster d'un gestionnaire de files d'attente de production existant seront prises en compte.](#)

### *Création des exemples de clusters*

Les définitions et les instructions permettant de créer l'exemple de cluster et de le modifier pour isoler la file d'attente SALES et séparer les messages sur le gestionnaire de files d'attente de passerelle.

## Pourquoi et quand exécuter cette tâche

Les commandes **MQSC** complètes permettant de créer les clusters FINANCE, SALES et Q.SALES sont fournies dans [Définitions pour les clusters de base](#), [Modifications visant à isoler la file d'attente de vente dans un nouveau cluster et à séparer les files d'attente de transmission du cluster de passerelle](#) et [Suppression de la file d'attente de vente sur le gestionnaire de files d'attente SALESRV du cluster de vente](#). Le cluster DEVELOP est omis dans les définitions, pour que les définitions soient plus courtes.

## Procédure

1. Créez les clusters SALES et FINANCE et le gestionnaire de files d'attente de passerelle.

a) Créez les gestionnaires de files d'attente.

Exécutez la commande `crtmqm -sax -u SYSTEM.DEAD.LETTER.QUEUE QmgrName` pour chacun des noms de gestionnaire de files d'attente dans Tableau 4, à la page 64.

Tableau 4. Noms de gestionnaire de files d'attente et numéros de port		
Description	Nom gest. de files	Numéro de port
Référentiel Finance	FINR1	1414
Référentiel Finance	FINR2	1415
Client financier	FINCLT	1418
Référentiel des ventes	SALER1	1416
Référentiel des ventes	SALER2	1417
Serveur de vente	SALESRV	1419
Passerelle	GATE	1420

b) Démarrer tous les gestionnaires de files d'attente

Exécutez la commande `strmqm QmgrName` pour chacun des noms de gestionnaire de files d'attente dans Tableau 4, à la page 64.

c) Création des définitions pour chacun des gestionnaires de files d'attente

Exécutez la commande: `runmqsc QmgrName <filename` où les fichiers sont répertoriés dans [Définitions pour les clusters de base](#), et le nom de fichier correspond au nom du gestionnaire de files d'attente.

### Définitions pour les clusters de base

#### **finr1.txt**

```
DEFINE LISTENER(1414) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1414) REPLACE
START LISTENER(1414)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSSDR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
```

#### **finr2.txt**

```
DEFINE LISTENER(1415) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1415) REPLACE
START LISTENER(1415)
ALTER QMGR REPOS(FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINR2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1415)')
CLUSTER(FINANCE) REPLACE
```

#### **finclt.txt**

```
DEFINE LISTENER(1418) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1418) REPLACE
START LISTENER(1418)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('localhost(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.FINCLT) CHLTYPE(CLUSRCVR) CONNAME('localhost(1418)')
CLUSTER(FINANCE) REPLACE
DEFINE QMODEL(SYSTEM.SAMPLE.REPLY) REPLACE
```



## saler1.txt

```
DEFINE LISTENER(1416) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1416) REPLACE
START LISTENER(1416)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
```

## saler2.txt

```
DEFINE LISTENER(1417) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1417) REPLACE
START LISTENER(1417)
ALTER QMGR REPOS(SALES)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(SALES) REPLACE
```

## salesrv.txt

```
DEFINE LISTENER(1419) TRPTYPE(TCP) IPADDR(localhost) CONTROL(QMGR) PORT(1419) REPLACE
START LISTENER(1419)
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.SALESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(SALES) REPLACE
DEFINE QLOCAL(SALES) CLUSTER(SALES) TRIGGER INITQ(SYSTEM.DEFAULT.INITIATION.QUEUE)
PROCESS(ECHO) REPLACE
DEFINE PROCESS(ECHO) APPLICID(AMQSECH) REPLACE
```

## gate.txt

```
DEFINE LISTENER(1420) TRPTYPE(TCP) IPADDR(LOCALHOST) CONTROL(QMGR) PORT(1420) REPLACE
START LISTENER(1420)
DEFINE NAMELIST(ALL) NAMES(SALES, FINANCE)
DEFINE CHANNEL(FINANCE.FINR1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1414)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(FINANCE.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(FINANCE) REPLACE
DEFINE CHANNEL(SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('LOCALHOST(1416)')
CLUSTER(SALES) REPLACE
DEFINE CHANNEL(SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('LOCALHOST(1420)')
CLUSTER(SALES) REPLACE
DEFINE QALIAS(A.SALES) CLUSNL(ALL) TARGET(SALES) TARGTYPE(Queue) DEFBIND(NOTFIXED)
REPLACE
DEFINE QREMOTE(FINCLT) RNAME(' ') RQMNAME(FINCLT) CLUSNL(ALL) REPLACE
DEFINE QREMOTE(SALESRV) RNAME(' ') RQMNAME(SALESRV) CLUSNL(ALL) REPLACE
```

2. Testez la configuration en exécutant l'exemple de programme de demande.

a) Démarrage du programme de moniteur de déclenchement sur le gestionnaire de files d'attente SALESRV

Sous Windows, ouvrez une fenêtre de commande et exécutez la commande `runmqtrm -m SALESRV`

b) Exécutez l'exemple de programme de demande et envoyez une demande.

Sous Windows, ouvrez une fenêtre de commande et exécutez la commande `amqsreq A.SALES FINCLT`

Le message de demande est renvoyé et, après 15 secondes, l'exemple de programme se termine.

3. Créez les définitions pour isoler la file d'attente SALES dans le cluster Q.SALES et séparer les messages de cluster pour les clusters SALES et FINANCE sur le gestionnaire de files d'attente de passerelle.

Exécutez la commande: `runmqsc QmgrName < filename` où les fichiers sont répertoriés dans la liste suivante et où le nom de fichier correspond presque au nom du gestionnaire de files d'attente.

## Modifications permettant d'isoler la file d'attente des ventes dans un nouveau cluster et de séparer les files d'attente de transmission du cluster de passerelle chgsaler1.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSSDR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSRCVR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
```

## chgsaler2.txt

```
DEFINE NAMELIST(CLUSTERS) NAMES(SALES, Q.SALES)
ALTER QMGR REPOS(' ') REPOSNL(CLUSTERS)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SALER2) CHLTYPE(CLUSRCVR) CONNAME('localhost(1417)')
CLUSTER(Q.SALES) REPLACE
```

## chgsalesrv.txt

```
DEFINE NAMELIST (CLUSTERS) NAMES(SALES, Q.SALES)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.SAVESRV) CHLTYPE(CLUSRCVR) CONNAME('localhost(1419)')
CLUSTER(Q.SALES) REPLACE
ALTER QLOCAL (SALES) CLUSTER(' ') CLUSNL(CLUSTERS)
```

## chgate.txt

```
ALTER NAMELIST(ALL) NAMES(SALES, FINANCE, Q.SALES)
ALTER QMGR DEFCLXQ(CHANNEL)
DEFINE CHANNEL(Q.SALES.SALER1) CHLTYPE(CLUSSDR) CONNAME('localhost(1416)')
CLUSTER(Q.SALES) REPLACE
DEFINE CHANNEL(Q.SALES.GATE) CHLTYPE(CLUSRCVR) CONNAME('localhost(1420)')
CLUSTER(Q.SALES) REPLACE
DEFINE QLOCAL (XMITQ.Q.SALES.SALESRV) USAGE(XMITQ) CLCHNAME(Q.SALES.SALESRV) REPLACE
DEFINE QLOCAL (XMITQ.SALES) USAGE(XMITQ) CLCHNAME(SALES.*) REPLACE
DEFINE QLOCAL (XMITQ.FINANCE) USAGE(XMITQ) CLCHNAME(FINANCE.*) REPLACE
```

4. Supprimez la file d'attente SALES du cluster SALES .

Exécutez la commande **MQSC** dans [Figure 15](#), à la page 66:

```
ALTER QLOCAL(SALES) CLUSTER('Q.SALES') CLUSNL(' ')
```

*Figure 15. Supprimez la file d'attente des ventes du gestionnaire de files d'attente SALESRV du cluster des ventes*

5. Basculez les canaux vers les nouvelles files d'attente de transmission.

Il est nécessaire d'arrêter et de démarrer tous les canaux utilisés par le gestionnaire de files d'attente GATE . Pour ce faire avec le moins de commandes, arrêtez et démarrez le gestionnaire de files d'attente

```
endmqm -i GATE
strmqm GATE
```

## Que faire ensuite

1. Réexécutez l'exemple de programme de demande pour vérifier que la nouvelle configuration fonctionne ; voir l'étape «2», à la page 65

2. Surveillez les messages transitant par toutes les files d'attente de transmission du cluster sur le gestionnaire de files d'attente GATE :
  - a. Modifiez la définition de chacune des files d'attente de transmission du cluster pour activer la surveillance des files d'attente.

```
ALTER QLOCAL(SYSTEM.CLUSTER.TRANSMIT.  
name) STATQ(ON)
```

- b. Vérifiez que la surveillance des statistiques du gestionnaire de files d'attente est OFF afin de réduire la sortie et définissez l'intervalle de surveillance sur une valeur inférieure pour effectuer plusieurs tests de manière pratique.

```
ALTER QMGR STATINT(60) STATCHL(OFF) STATQ(OFF) STATMQI(OFF) STATACLS(OFF)
```

- c. Redémarrez le gestionnaire de files d'attente GATE .
  - d. Exécutez l'exemple de programme de demande plusieurs fois pour vérifier qu'un nombre égal de messages transitent par SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV et SYSTEM.CLUSTER.TRANSMIT.QUEUE. Les demandes passent par SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SALESRV et les réponses par SYSTEM.CLUSTER.TRANSMIT.QUEUE.

```
amqsmon -m GATE -t statistics
```

- e. Les résultats sur quelques intervalles sont les suivants:

```
C:\Documents and Settings\Admin>amqsmon -m GATE -t statistics
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '14.59.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.00.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
GetCount: [1, 0]
GetBytes: [435, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [1, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [435, 0]
```

```

GetCount: [1, 0]
GetBytes: [435, 0]
...
MonitoringType: QueueStatistics
QueueManager: 'GATE'
IntervalStartDate: '2012-02-27'
IntervalStartTime: '15.00.20'
IntervalEndDate: '2012-02-27'
IntervalEndTime: '15.01.20'
CommandLevel: 700
ObjectCount: 2
QueueStatistics: 0
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.QUEUE'
CreateDate: '2012-02-24'
CreateTime: '15.58.15'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
QueueStatistics: 1
QueueName: 'SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV'
CreateDate: '2012-02-24'
CreateTime: '16.37.43'
...
PutCount: [2, 0]
PutFailCount: 0
Put1Count: [0, 0]
Put1FailCount: 0
PutBytes: [863, 0]
GetCount: [2, 0]
GetBytes: [863, 0]
...
2 Records Processed.

```

Un message de demande et de réponse a été envoyé dans le premier intervalle et deux dans le second. Vous pouvez déduire que les messages de demande ont été placés sur SYSTEM.CLUSTER.TRANSMIT.Q.SALES.SAVESRV et les messages de réponse sur SYSTEM.CLUSTER.TRANSMIT.QUEUE.

*Mise en cluster: changement des files d'attente de transmission de cluster*

Planifiez la manière dont les modifications apportées aux files d'attente de transmission de cluster d'un gestionnaire de files d'attente de production existant seront prises en compte.

## Avant de commencer

Si vous réduisez le nombre de messages que le processus de commutation doit transférer vers la nouvelle file d'attente de transmission, la commutation se termine plus rapidement. Lisez la rubrique [Fonctionnement du processus de basculement du canal émetteur de cluster vers une autre file d'attente de transmission](#) pour connaître les raisons de la tentative de vidage de la file d'attente de transmission avant de poursuivre.

## Pourquoi et quand exécuter cette tâche

Vous avez le choix entre deux méthodes pour appliquer les modifications aux files d'attente de transmission de cluster.

1. Laissez le gestionnaire de files d'attente effectuer les modifications automatiquement. Il s'agit de l'option par défaut. Le gestionnaire de files d'attente bascule les canaux émetteurs de cluster avec des modifications de file d'attente de transmission en attente lors du prochain démarrage d'un canal émetteur de cluster.
2. Effectuez les modifications manuellement. Vous pouvez apporter les modifications à un canal émetteur de cluster lorsqu'il est arrêté. Vous pouvez le basculer d'une file d'attente de transmission de cluster à une autre avant le démarrage du canal émetteur de cluster.

Quels sont les facteurs que vous prenez en compte lorsque vous décidez laquelle des deux options à choisir et comment gérez-vous le commutateur?

## Procédure

- Option 1: Laissez le gestionnaire de files d'attente effectuer les modifications automatiquement ; voir «Basculement des canaux émetteurs de cluster actifs vers un autre ensemble de files d'attente de transmission de cluster», à la page 70.

Choisissez cette option si vous souhaitez que le gestionnaire de files d'attente fasse le changement pour vous.

Une autre manière de décrire cette option consiste à dire que le gestionnaire de files d'attente bascule un canal émetteur de cluster sans que vous ne forçiez l'arrêt du canal. Vous avez la possibilité de forcer le canal à s'arrêter, puis de le démarrer, pour que le commutateur se produise plus tôt. Le commutateur démarre lorsque le canal démarre et s'exécute lorsque le canal est en cours d'exécution, ce qui est différent de l'option 2. Dans l'option 2, le commutateur a lieu lorsque le canal est arrêté.

Si vous choisissez cette option en laissant le commutateur se produire automatiquement, le processus de commutation démarre lorsqu'un canal émetteur de cluster démarre. Si le canal n'est pas arrêté, il démarre une fois qu'il devient inactif, s'il y a un message à traiter. Si le canal est arrêté, démarrez-le à l'aide de la commande `START CHANNEL`.

Le processus de commutation se termine dès qu'il ne reste plus de messages pour le canal émetteur de cluster dans la file d'attente de transmission que le canal servait. Dès que c'est le cas, les messages nouvellement arrivés pour le canal émetteur de cluster sont stockés directement dans la nouvelle file d'attente de transmission. Jusque-là, les messages sont stockés dans l'ancienne file d'attente de transmission, et le processus de commutation transfère les messages de l'ancienne file d'attente de transmission vers la nouvelle file d'attente de transmission. Le canal émetteur de cluster transmet les messages de la nouvelle file d'attente de transmission de cluster pendant tout le processus de commutation.

La fin du processus de commutation dépend de l'état du système. Si vous effectuez les modifications dans une fenêtre de maintenance, déterminez à l'avance si le processus de changement se termine à temps. Son exécution dans le temps dépend du fait que le nombre de messages en attente de transfert à partir de l'ancienne file d'attente de transmission atteint zéro.

L'avantage de la première méthode est qu'elle est automatique. Un inconvénient est que si le temps nécessaire pour effectuer les modifications de configuration est limité à une fenêtre de maintenance, vous devez être sûr de pouvoir contrôler le système pour terminer le processus de commutation dans la fenêtre de maintenance. Si vous ne pouvez pas en être sûr, l'option 2 peut être un meilleur choix.

- Option 2: Effectuez les modifications manuellement ; voir «Basculement d'un canal émetteur de cluster arrêté vers une autre file d'attente de transmission de cluster», à la page 71.

Choisissez cette option si vous souhaitez contrôler l'ensemble du processus de commutation manuellement ou si vous souhaitez changer de canal arrêté ou inactif. C'est un bon choix, si vous permutez quelques canaux émetteurs de cluster et que vous souhaitez effectuer le basculement lors d'une fenêtre de maintenance.

Une autre description de cette option consiste à indiquer que vous changez de canal émetteur de cluster, alors que le canal émetteur de cluster est arrêté.

Si vous choisissez cette option, vous avez un contrôle total sur le moment où le commutateur a lieu. Vous pouvez être certain d'avoir terminé le processus de commutation dans un laps de temps fixe, dans une fenêtre de maintenance. Le temps que prend le commutateur dépend du nombre de messages à transférer d'une file d'attente de transmission à l'autre. Si les messages continuent d'arriver, le processus peut prendre un certain temps pour transférer tous les messages.

Vous avez la possibilité de changer de canal sans transférer les messages de l'ancienne file d'attente de transmission. Le commutateur est "instantané".

Lorsque vous redémarrez le canal émetteur de cluster, il commence à traiter les messages de la file d'attente de transmission que vous venez d'affecter.

L'avantage de la deuxième méthode est que vous avez le contrôle sur le processus de commutation. L'inconvénient est que vous devez identifier les canaux émetteurs de cluster à commuter, exécuter les commandes nécessaires et résoudre les canaux en attente de validation qui pourraient empêcher l'arrêt du canal émetteur de cluster.

### Concepts associés

[Comment choisir le type de file d'attente de transmission de cluster à utiliser](#)

[Comment choisir entre différentes options de configuration de file d'attente de transmission de cluster.](#)

[Fonctionnement du processus de basculement du canal émetteur de cluster vers une autre file d'attente de transmission](#)

### Tâches associées

[Mise en cluster: exemple de configuration de plusieurs files d'attente de transmission de cluster](#)

Dans cette tâche, vous appliquez les étapes de planification de plusieurs files d'attente de transmission de cluster à trois clusters qui se chevauchent. Les exigences sont de séparer les flux de messages vers une file d'attente de cluster, à partir de tous les autres flux de messages, et de stocker les messages de différents clusters dans différentes files d'attente de transmission de cluster.

*Basculement des canaux émetteurs de cluster actifs vers un autre ensemble de files d'attente de transmission de cluster*

Cette tâche vous offre trois options pour la commutation des canaux émetteurs de cluster actifs. Une option consiste à laisser le gestionnaire de files d'attente effectuer le basculement automatiquement, ce qui n'affecte pas les applications en cours d'exécution. Les autres options permettent d'arrêter et de démarrer les canaux manuellement ou de redémarrer le gestionnaire de files d'attente.

### Avant de commencer

Modifiez la configuration de la file d'attente de transmission du cluster. Vous pouvez modifier l'attribut de gestionnaire de files d'attente **DEFCLXQ** ou ajouter ou modifier l'attribut **CLCHNAME** des files d'attente de transmission.

Si vous réduisez le nombre de messages que le processus de commutation doit transférer vers la nouvelle file d'attente de transmission, la commutation se termine plus rapidement. Lisez la rubrique [Fonctionnement du processus de basculement du canal émetteur de cluster vers une autre file d'attente de transmission](#) pour connaître les raisons de la tentative de vidage de la file d'attente de transmission avant de poursuivre.

### Pourquoi et quand exécuter cette tâche

Utilisez les étapes de la tâche comme base pour l'élaboration de votre propre plan de modification de la configuration de la file d'attente de transmission de cluster.

### Procédure

1. Facultatif : Enregistrer le statut en cours du canal

Enregistrez le statut des canaux en cours et sauvegardés qui servent les files d'attente de transmission du cluster. Les commandes suivantes affichent l'état associé aux files d'attente de transmission de grappe du système. Ajoutez vos propres commandes pour afficher l'état associé aux files d'attente de transmission de cluster que vous avez définies. Utilisez une convention, telle que XMITQ. *ChannelName*, pour nommer les files d'attente de transmission de cluster que vous définissez afin de faciliter l'affichage de l'état du canal pour ces files d'attente de transmission.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
```

## 2. Commuter les files d'attente de transmission.

- Aucune action requise. Le gestionnaire de files d'attente bascule les canaux émetteurs de cluster lorsqu'ils redémarrent après avoir été arrêtés ou inactifs.

Choisissez cette option si vous n'avez pas de règles ou si vous souhaitez modifier la configuration d'un gestionnaire de files d'attente. Les applications en cours d'exécution ne sont pas affectées par les modifications.

- Redémarrez le gestionnaire de files d'attente. Tous les canaux émetteurs de cluster sont arrêtés et redémarrés automatiquement à la demande.

Choisissez cette option pour initier immédiatement toutes les modifications. Les applications en cours d'exécution sont interrompues par le gestionnaire de files d'attente à mesure qu'il s'arrête et redémarre.

- Arrêtez les canaux émetteurs de cluster individuels et redémarrez-les.

Choisissez cette option pour modifier immédiatement quelques canaux. Les applications en cours d'exécution subissent un court délai de transfert de message entre l'arrêt et le redémarrage du canal de transmission de messages. Le canal émetteur de cluster reste en cours d'exécution, sauf lorsque vous l'avez arrêté. Au cours du processus de commutation, les messages sont distribués à l'ancienne file d'attente de transmission, transférés à la nouvelle file d'attente de transmission par le processus de commutation et transmis à partir de la nouvelle file d'attente de transmission par le canal émetteur de cluster.

## 3. Facultatif : Surveiller les canaux au fur et à mesure qu'ils changent

Affichez le statut du canal et la longueur de la file d'attente de transmission lors du basculement. L'exemple suivant affiche l'état des files d'attente de transmission de cluster système.

```
DISPLAY CHSTATUS(*) WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY CHSTATUS(*) SAVED WHERE(XMITQ LK 'SYSTEM.CLUSTER.TRANSMIT.*')
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

## 4. Facultatif : Surveillez les messages AMQ7341 La file d'attente de transmission du canal *ChannelName* est passée de la file d'attente *QueueName* à la file d'attente *QueueName* qui sont écrits dans le journal des erreurs du gestionnaire de files d'attente.

*Basculement d'un canal émetteur de cluster arrêté vers une autre file d'attente de transmission de cluster*  
Si vous choisissez d'apporter des modifications manuellement, vous devez apporter les modifications à un canal émetteur de cluster lorsqu'il est arrêté et le faire passer d'une file d'attente de transmission de cluster à une autre avant le démarrage du canal émetteur de cluster.

## Avant de commencer

Vous pouvez apporter des modifications à la configuration et souhaitez désormais les rendre effectives sans démarrer les canaux émetteurs de cluster qui sont affectés. Vous pouvez également effectuer les modifications de configuration dont vous avez besoin lors de l'une des étapes de la tâche.

Si vous réduisez le nombre de messages que le processus de commutation doit transférer vers la nouvelle file d'attente de transmission, la commutation se termine plus rapidement. Lisez la rubrique [Fonctionnement du processus de basculement du canal émetteur de cluster vers une autre file d'attente](#)

de transmission pour connaître les raisons de la tentative de vidage de la file d'attente de transmission avant de poursuivre.

## Pourquoi et quand exécuter cette tâche

Cette tâche permute les files d'attente de transmission servies par les canaux émetteurs de cluster arrêtés ou inactifs. Vous pouvez effectuer cette tâche car un canal émetteur de cluster est arrêté et vous souhaitez changer immédiatement de file d'attente de transmission. Par exemple, pour une raison quelconque, un canal émetteur de cluster ne démarre pas ou a un autre problème de configuration. Pour résoudre le problème, vous décidez de créer un canal émetteur de cluster et d'associer la file d'attente de transmission de l'ancien canal émetteur de cluster au nouveau canal émetteur de cluster que vous avez défini.

Il est plus probable que vous souhaitiez contrôler le moment où la reconfiguration des files d'attente de transmission du cluster est effectuée. Pour contrôler complètement la reconfiguration, vous arrêtez les canaux, modifiez la configuration, puis changez les files d'attente de transmission.

## Procédure

1. Arrêtez les canaux que vous avez l'intention de changer

- a) Arrêtez les canaux en cours d'exécution ou inactifs que vous prévoyez de permuter. L'arrêt d'un canal émetteur de cluster inactif l'empêche de démarrer pendant que vous apportez des modifications à la configuration.

```
STOP CHANNEL(ChannelName) MODE(QUIESCSE) STATUS(STOPPED)
```


2. Facultatif : Effectuez les modifications de configuration.

Pour un exemple, voir [«Mise en cluster: exemple de configuration de plusieurs files d'attente de transmission de cluster»](#), à la page 60.

3. Basculez les canaux émetteurs de cluster vers les nouvelles files d'attente de transmission de cluster.

 Sous [Multiplateformes](#), exécutez la commande suivante:

```
runswchl -m QmgrName -c ChannelName
```

 Sous z/OS, utilisez la fonction SWITCH de la commande CSQUTIL pour changer les messages ou surveiller ce qui se passe. Utilisez la commande suivante.

```
SWITCH CHANNEL(channel_name) MOVEMSGS(YES)
```

Pour plus d'informations, voir [Fonction SWITCH](#).

La commande **runswchl**, ou CSQUTIL SWITCH, transfère tous les messages de l'ancienne file d'attente de transmission vers la nouvelle file d'attente de transmission. Lorsque le nombre de messages de l'ancienne file d'attente de transmission de ce canal atteint zéro, le commutateur est terminé. La commande est synchrone. La commande écrit des messages de progression dans la fenêtre pendant le processus de commutation.

Lors de la phase de transfert, les messages existants et nouveaux destinés au canal émetteur de cluster sont transférés vers la nouvelle file d'attente de transmission.

Le canal émetteur de cluster étant arrêté, les messages s'accumulent dans la nouvelle file d'attente de transmission. Comparez le canal émetteur de cluster arrêté à l'étape «2», à la page 71 dans [«Basculement des canaux émetteurs de cluster actifs vers un autre ensemble de files d'attente de transmission de cluster»](#), à la page 70. Dans cette étape, le canal émetteur de cluster est en cours d'exécution, de sorte que les messages ne s'accumulent pas nécessairement dans la nouvelle file d'attente de transmission.



#### 4. Facultatif : Surveiller les canaux au fur et à mesure qu'ils changent

Dans une autre fenêtre de commande, affichez la longueur de la file d'attente de transmission lors du basculement. L'exemple suivant affiche l'état des files d'attente de transmission de cluster système.

```
DISPLAY QUEUE('SYSTEM.CLUSTER.TRANSMIT.*') CURDEPTH
```

5. Facultatif : Surveillez les messages AMQ7341 La file d'attente de transmission du canal *ChannelName* est passée de la file d'attente *QueueName* à la file d'attente *QueueName* qui sont écrits dans le journal des erreurs du gestionnaire de files d'attente.

6. Redémarrez les canaux émetteurs de cluster que vous avez arrêtés.

Les canaux ne démarrent pas automatiquement, car vous les avez arrêtés, en les plaçant à l'état ARRETE .

```
START CHANNEL(ChannelName)
```

### Référence associée

[runswchl](#)

[Résolution du canal](#)

[Arrêter le canal](#)

*Mise en cluster: meilleures pratiques en matière de migration et de modification*

Cette rubrique fournit des conseils pour la planification et l'administration des clusters IBM MQ . Ces informations sont un guide basé sur les tests et les commentaires des clients.

1. «Déplacement d'objets dans un cluster», à la page 73 (pratiques recommandées pour le déplacement d'objets à l'intérieur d'un cluster, sans installer de groupes de correctifs ou de nouvelles versions d' IBM MQ ).
2. «Mises à niveau et installations de maintenance», à la page 75 (meilleures pratiques pour maintenir une architecture de cluster opérationnelle, tout en appliquant des opérations de maintenance ou de mise à niveau et en testant la nouvelle architecture).

## Déplacement d'objets dans un cluster

### Applications et leurs files d'attente

Lorsque vous devez déplacer une instance de file d'attente hébergée sur un gestionnaire de files d'attente pour qu'elle soit hébergée sur un autre, vous pouvez utiliser les paramètres d'équilibrage de charge pour assurer une transition en douceur.

Créez une instance de la file d'attente dans laquelle elle doit être nouvellement hébergée, mais utilisez les paramètres d'équilibrage de charge de cluster pour continuer à envoyer des messages à l'instance d'origine jusqu'à ce que votre application soit prête à basculer. Pour ce faire, procédez comme suit:

1. Définissez la propriété **CLWLRANK** de la file d'attente existante sur une valeur élevée, par exemple cinq.
2. Créez la nouvelle instance de la file d'attente et définissez sa propriété **CLWLRANK** sur zéro.
3. Effectuez toute configuration supplémentaire du nouveau système, par exemple, déployez et commencez à utiliser des applications sur la nouvelle instance de la file d'attente.
4. Définissez la propriété **CLWLRANK** de la nouvelle instance de file d'attente de sorte qu'elle soit supérieure à l'instance d'origine, par exemple neuf.
5. Autorisez l'instance de file d'attente d'origine à traiter tous les messages mis en file d'attente dans le système, puis supprimez la file d'attente.

## Déplacement de gestionnaires de files d'attente entiers

Si le gestionnaire de files d'attente reste sur le même hôte, mais que l'adresse IP est modifiée, le processus est le suivant:

- Le DNS, lorsqu'il est utilisé correctement, peut aider à simplifier le processus. Pour plus d'informations sur l'utilisation du DNS en définissant l'attribut de canal Nom de connexion (CONNAME), voir ALTER CHANNEL.
- Si vous déplacez un référentiel complet, vérifiez que vous disposez d'au moins un autre référentiel complet qui s'exécute correctement (aucun problème avec le statut du canal, par exemple) avant d'apporter des modifications.
- Suspendez le gestionnaire de files d'attente à l'aide de la commande SUSPEND QMGR pour éviter l'accumulation de trafic.
- Modifiez l'adresse IP de l'ordinateur. Si votre définition de canal CLUSRCVR utilise une adresse IP dans la zone CONNAME, modifiez cette entrée d'adresse IP. Il peut être nécessaire de vider le cache DNS pour s'assurer que les mises à jour sont disponibles partout.
- Lorsque le gestionnaire de files d'attente se reconnecte aux référentiels complets, les définitions automatiques de canal se résolvent automatiquement.
- Si le gestionnaire de files d'attente a hébergé un référentiel complet et que l'adresse IP change, il est important de s'assurer que les partitions sont basculées dès que possible pour pointer les canaux CLUSSDR définis manuellement vers le nouvel emplacement. Tant que ce basculement n'est pas effectué, ces gestionnaires de files d'attente peuvent être en mesure de contacter uniquement le référentiel complet restant (inchangé) et des messages d'avertissement peuvent être affichés concernant la définition de canal incorrecte.
- Reprenez le gestionnaire de files d'attente à l'aide de la commande RESUME QMGR.

Si le gestionnaire de files d'attente doit être déplacé vers un nouvel hôte, il est possible de copier les données du gestionnaire de files d'attente et de les restaurer à partir d'une sauvegarde. Ce processus n'est toutefois pas recommandé, sauf s'il n'existe pas d'autres options ; il peut être préférable de créer un gestionnaire de files d'attente sur une nouvelle machine et de répliquer les files d'attente et les applications, comme décrit dans la section précédente. Cette situation donne un mécanisme de remise à zéro / annulation en douceur.

Si vous êtes déterminé à déplacer un gestionnaire de files d'attente complet à l'aide de la sauvegarde, suivez les meilleures pratiques suivantes:

- Traitez l'ensemble du processus comme une restauration de gestionnaire de files d'attente à partir d'une sauvegarde, en appliquant tous les processus que vous utiliseriez généralement pour la récupération du système en fonction de votre environnement de système d'exploitation.
- Utilisez la commande **REFRESH CLUSTER** après la migration pour supprimer toutes les informations de cluster détenues localement (y compris les canaux définis automatiquement qui sont en attente de validation) et forcer leur régénération.

**Remarque :** Pour les clusters de grande taille, l'utilisation de la commande **REFRESH CLUSTER** peut perturber le cluster lors de son exécution, ainsi qu'ensuite, à des intervalles de 27 jours, lorsque les objets du cluster envoient automatiquement des mises à jour de statut à tous les gestionnaires de files d'attente intéressés. Voir L'actualisation d'un grand cluster peut affecter les performances et la disponibilité du cluster.

Lors de la création d'un gestionnaire de files d'attente et de la réplification de la configuration à partir d'un gestionnaire de files d'attente existant dans le cluster (comme décrit précédemment dans cette rubrique), ne traitez jamais les deux gestionnaires de files d'attente différents comme étant identiques. En particulier, n'attribuez pas à un nouveau gestionnaire de files d'attente le même nom de gestionnaire de files d'attente et la même adresse IP. La tentative de suppression d'un gestionnaire de files d'attente de remplacement est une cause fréquente de problèmes dans les clusters IBM MQ. Le cache s'attend à recevoir des mises à jour, y compris l'attribut **QMID**, et l'état peut être endommagé.

Si deux gestionnaires de files d'attente différents sont créés accidentellement avec le même nom, il est recommandé d'utiliser la commande **RESET CLUSTER QMID** pour éjecter l'entrée incorrecte du cluster.

## Mises à niveau et installations de maintenance

Évitez le scénario dit big bang (par exemple, l'arrêt de toutes les activités de cluster et de gestionnaire de files d'attente, l'application de toutes les mises à niveau et de la maintenance à tous les gestionnaires de files d'attente, puis le démarrage de tout en même temps). Les clusters étant conçus pour continuer à fonctionner avec plusieurs versions du gestionnaire de files d'attente coexistant, une approche de maintenance par étapes bien planifiée est recommandée.

Disposer d'un plan de sauvegarde:

- Avez-vous effectué des sauvegardes?
- Évitez d'utiliser une nouvelle fonctionnalité de cluster immédiatement: attendez que vous soyez certain que tous les gestionnaires de files d'attente sont mis à niveau vers le nouveau niveau et que vous soyez certain que vous ne les annulez pas. L'utilisation d'une nouvelle fonction de cluster dans un cluster où certains gestionnaires de files d'attente sont encore à un niveau antérieur peut entraîner un comportement non défini.

Un référentiel stocke l'enregistrement qu'il reçoit dans sa propre version. Si l'enregistrement reçu est d'une version ultérieure, les attributs de la version ultérieure sont supprimés lorsque l'enregistrement est stocké. Un gestionnaire de files d'attente IBM MQ 9.3 qui reçoit des informations relatives à un gestionnaire de files d'attente de IBM MQ 9.4 stocke uniquement les informations relatives à IBM MQ 9.3. Un référentiel de IBM MQ 9.4 qui reçoit un enregistrement de IBM MQ 9.3 stocke les valeurs par défaut des attributs introduits dans la version plus récente. Les valeurs par défaut définissent les valeurs des attributs qui ne sont pas inclus dans l'enregistrement reçu.

Migrez d'abord les référentiels complets. Bien qu'ils puissent transmettre des informations qu'ils ne comprennent pas, ils ne peuvent pas les conserver, ce n'est donc pas l'approche recommandée à moins qu'elle ne soit absolument nécessaire. Pour plus d'informations, voir [Migration de cluster de gestionnaires de files d'attente](#).

### *Mise en cluster : meilleures pratiques d'utilisation REFRESH CLUSTER*

Vous utilisez la commande **REFRESH CLUSTER** pour supprimer toutes les informations stockées localement sur un cluster et régénérer ces informations à partir des référentiels complets du cluster. Vous ne devez pas utiliser cette commande, sauf dans des circonstances exceptionnelles. Si vous avez besoin de l'utiliser, il y a des considérations spéciales sur la façon dont vous l'utilisez. Ces informations sont un guide basé sur les tests et les commentaires des clients.

## N'exécutez REFRESH CLUSTER que si vous en avez vraiment besoin

La technologie de cluster IBM MQ garantit que toute modification apportée à la configuration du cluster, telle qu'une modification apportée à une file d'attente de cluster, est automatiquement reconnue par tout membre du cluster qui a besoin de connaître les informations. Il n'est pas nécessaire de prendre d'autres mesures administratives pour parvenir à cette propagation de l'information.

Si ces informations n'atteignent pas les gestionnaires de files d'attente du cluster où elles sont requises, par exemple si une file d'attente en cluster n'est pas connue d'un autre gestionnaire de files d'attente du cluster lorsqu'une application tente de l'ouvrir pour la première fois, cela implique un problème dans l'infrastructure du cluster. Par exemple, il est possible qu'un canal ne puisse pas être démarré entre un gestionnaire de files d'attente et un gestionnaire de files d'attente de référentiel complet. Par conséquent, toute situation dans laquelle des incohérences sont observées doit faire l'objet d'une enquête. Si possible, résolvez la situation sans utiliser la commande **REFRESH CLUSTER**.

Dans de rares cas documentés ailleurs dans la documentation de ce produit, ou à la demande du support IBM, vous pouvez utiliser la commande **REFRESH CLUSTER** pour supprimer toutes les informations détenues en local sur un cluster et régénérer ces informations à partir des référentiels complets du cluster.

## L'actualisation dans un cluster de grande taille peut affecter les performances et la disponibilité du cluster

L'utilisation de la commande **REFRESH CLUSTER** peut perturber le cluster pendant qu'il est en cours, par exemple en créant une augmentation soudaine du travail pour les référentiels complets lorsqu'ils traitent la repropagation des ressources de cluster du gestionnaire de files d'attente. Si vous effectuez une régénération dans un grand cluster (c'est-à-dire plusieurs centaines de gestionnaires de files d'attente), vous devez éviter d'utiliser la commande dans le travail quotidien si possible et utiliser d'autres méthodes pour corriger des incohérences spécifiques. Par exemple, si une file d'attente de cluster n'est pas correctement propagée dans le cluster, une technique d'investigation initiale consistant à mettre à jour la définition de file d'attente de cluster, telle que la modification de sa description, repropage la configuration de file d'attente dans le cluster. Ce processus peut aider à identifier le problème et à résoudre éventuellement une incohérence temporaire.

Si d'autres méthodes ne peuvent pas être utilisées et que vous devez exécuter **REFRESH CLUSTER** dans un cluster de grande taille, vous devez le faire aux heures creuses ou pendant une fenêtre de maintenance afin d'éviter tout impact sur les charges de travail des utilisateurs. Vous devez également éviter d'actualiser un grand cluster dans un seul lot et d'échelonner l'activité comme expliqué dans [«Éviter les problèmes de performances et de disponibilité lorsque les objets de cluster envoient des mises à jour automatiques»](#), à la page 76.

## Éviter les problèmes de performances et de disponibilité lorsque les objets de cluster envoient des mises à jour automatiques

Une fois qu'un nouvel objet de cluster est défini sur un gestionnaire de files d'attente, une mise à jour de cet objet est générée tous les 27 jours à partir de la définition et envoyée à chaque référentiel complet du cluster et à tous les autres gestionnaires de files d'attente intéressés. Lorsque vous exécutez la commande **REFRESH CLUSTER** sur un gestionnaire de files d'attente, vous réinitialisez l'horloge de cette mise à jour automatique sur tous les objets définis localement dans le cluster spécifié.

Si vous régénérez un cluster de grande taille (c'est-à-dire plusieurs centaines de gestionnaires de files d'attente) en un seul lot ou dans d'autres circonstances, par exemple en recréant un système à partir d'une sauvegarde de configuration, au bout de 27 jours, tous ces gestionnaires de files d'attente réannoncent toutes leurs définitions d'objet dans les référentiels complets en même temps. Cela peut à nouveau entraîner une exécution beaucoup plus lente du système, voire une indisponibilité, jusqu'à ce que toutes les mises à jour soient terminées. Par conséquent, lorsque vous devez actualiser ou recréer plusieurs gestionnaires de files d'attente dans un cluster de grande taille, vous devez échelonner l'activité sur plusieurs heures ou plusieurs jours afin que les mises à jour automatiques ultérieures n'aient pas d'impact régulier sur les performances du système.

## File d'attente de l'historique du cluster du système

Lorsqu'une **REFRESH CLUSTER** est exécutée, le gestionnaire de files d'attente prend une image instantanée de l'état du cluster avant l'actualisation et le stocke sur le `SYSTEM.CLUSTER.HISTORY.QUEUE (SCHQ)` s'il est défini sur le gestionnaire de files d'attente. Cet instantané est destiné au service IBM uniquement, en cas de problèmes ultérieurs avec le système.

Le SCHQ est défini par défaut sur les gestionnaires de files d'attente réparties au démarrage. Pour la migration z/OS, le SCHQ doit être défini manuellement.

Les messages du SCHQ expirent au bout de trois mois.

### Concepts associés

[«Considérations relatives à l'ACTUALISATION DU CLUSTER pour les clusters de publication / abonnement»](#), à la page 114

L'émission de la commande **REFRESH CLUSTER** entraîne la suppression temporaire par le gestionnaire de files d'attente des informations locales sur un cluster, y compris les rubriques de cluster et les abonnements de proxy associés.

## Référence associée

Problèmes d'application rencontrés lors de l'exécution de [REFRESH CLUSTER](#)

Référence des commandes MQSC: [REFRESH CLUSTER](#)

*Mise en cluster: disponibilité, multi-instance et reprise après incident*

Cette rubrique fournit des conseils pour la planification et l'administration des clusters IBM MQ . Ces informations sont un guide basé sur les tests et les commentaires des clients.

IBM MQ La mise en cluster elle-même n'est pas une solution à haute disponibilité, mais dans certains cas, elle peut être utilisée pour améliorer la disponibilité des services à l'aide de IBM MQ, par exemple en ayant plusieurs instances d'une file d'attente sur différents gestionnaires de files d'attente. La présente section explique comment s'assurer que l'infrastructure IBM MQ est aussi hautement disponible que possible afin qu'elle puisse être utilisée dans une telle architecture.

**Remarque :** D'autres solutions de haute disponibilité et de reprise après incident sont disponibles pour IBM MQ. Voir [Configuration de la haute disponibilité, de la reprise et du redémarrage](#).

### Disponibilité des ressources de cluster

La raison de la recommandation habituelle de maintenir deux référentiels complets est que la perte d'un d'entre eux n'est pas critique pour le bon fonctionnement du cluster. Même si les deux deviennent indisponibles, il existe un délai de grâce de 60 jours pour les connaissances existantes détenues par des référentiels partiels, bien que des ressources nouvelles ou non précédemment consultées (par exemple, des files d'attente) ne soient pas disponibles dans cet événement.

### Utilisation de clusters pour améliorer la disponibilité des applications

Un cluster peut vous aider à concevoir des applications à haute disponibilité (par exemple, une application serveur de type demande / réponse), en utilisant plusieurs instances de la file d'attente et de l'application. Si nécessaire, les attributs de priorité peuvent donner la préférence à l'application "active", sauf si un gestionnaire de files d'attente ou un canal, par exemple, devient indisponible. Cela permet de basculer rapidement pour continuer à traiter les nouveaux messages lorsqu'un problème se produit.

Toutefois, les messages qui ont été distribués à un gestionnaire de files d'attente particulier dans un cluster sont conservés uniquement sur cette instance de file d'attente et ne peuvent pas être traités tant que ce gestionnaire de files d'attente n'a pas été récupéré. Pour cette raison, pour une haute disponibilité des données réelle, vous pouvez envisager d'autres technologies telles que les gestionnaires de files d'attente multi-instance.

### Gestionnaires de files d'attente multi-instance


La haute disponibilité des logiciels (multi-instance) est une offre intégrée permettant de conserver vos messages existants à disposition. Pour plus d'informations, voir [Utilisation d' IBM MQ avec des configurations à haute disponibilité, Création d'un gestionnaire de files d'attente multi-instance](#) et la section suivante. Tout gestionnaire de files d'attente d'un cluster peut être rendu hautement disponible à l'aide de cette technique, à condition que tous les gestionnaires de files d'attente du cluster exécutent au moins IBM WebSphere MQ 7.0.1. Si des gestionnaires de files d'attente du cluster se trouvent à des niveaux antérieurs, ils risquent de perdre la connectivité avec les gestionnaires de files d'attente multi-instance s'ils basculent sur une adresse IP secondaire.

Comme indiqué précédemment dans cette rubrique, tant que deux référentiels complets sont configurés, ils sont presque de par leur nature hautement disponibles. Si nécessaire, les gestionnaires de files d'attente IBM MQ Software High Availability / multi-instance peuvent être utilisés pour les référentiels complets. Il n'y a pas de raison forte d'utiliser ces méthodes et, en fait, pour les indisponibilités temporaires, ces méthodes peuvent entraîner des coûts de performances supplémentaires lors de la reprise en ligne. L'utilisation de la haute disponibilité des logiciels au lieu de l'exécution de deux référentiels complets est déconseillée car, par exemple, en cas d'indisponibilité d'un canal unique, elle ne basculerait pas nécessairement, mais pourrait laisser des référentiels partiels incapables de demander des ressources de cluster.

### Reprise après incident

La reprise après incident, par exemple lorsque les disques stockant les données d'un gestionnaire de files d'attente sont endommagés, est difficile à effectuer correctement ; IBM MQ peut vous aider, mais il ne peut pas le faire automatiquement. La seule option de reprise après incident>true'dans IBM MQ (à

l'exclusion de tout système d'exploitation ou d'autres technologies de réplication sous-jacentes) est la restauration à partir d'une sauvegarde. Il existe des points spécifiques au cluster à prendre en compte dans les situations suivantes:

- Faites attention lorsque vous testez des scénarios de reprise après incident. Par exemple, si vous testez le fonctionnement des gestionnaires de files d'attente de sauvegarde, soyez prudent lorsque vous les mettez en ligne dans le même réseau car il est possible de rejoindre accidentellement le cluster actif et de commencer à "voler" des messages en hébergeant les mêmes files d'attente nommées que dans les gestionnaires de files d'attente de cluster actif.
- Les tests de reprise après incident ne doivent pas interférer avec un cluster actif en cours d'exécution. Les techniques permettant d'éviter les interférences sont les suivantes:
  - Séparation complète du réseau ou séparation au niveau du pare-feu.
  -  Pas de démarrage du canal ou de l'espace adresse z/OS **chinit** .
  - Ne pas émettre de certificat TLS actif sur le système de reprise après incident jusqu'à ce qu'un scénario de reprise après incident réel se produise ou à moins qu'il ne se produise.
- Lors de la restauration d'une sauvegarde d'un gestionnaire de files d'attente dans le cluster, il est possible que la sauvegarde ne soit pas synchronisée avec le reste du cluster. La commande **REFRESH CLUSTER** peut résoudre les mises à jour et se synchroniser avec le cluster, mais la commande **REFRESH CLUSTER** doit être utilisée en dernier ressort. Voir «[Mise en cluster : meilleures pratiques d'utilisation REFRESH CLUSTER](#)», à la page 75. Consultez la documentation des processus internes et la documentation IBM MQ pour voir si une étape simple a été omise avant d'utiliser la commande.
- Comme pour toute reprise, les applications doivent gérer la relecture et la perte de données. Il faut décider s'il faut supprimer les files d'attente jusqu'à un état connu, ou s'il y a suffisamment d'informations ailleurs pour gérer les réexecutions.

## Planification de votre réseau de publication / abonnement distribué

Vous pouvez créer un réseau de gestionnaires de files d'attente dans lequel les abonnements créés sur un gestionnaire de files d'attente recevront des messages correspondants publiés par une application connectée à un autre gestionnaire de files d'attente du réseau. Pour choisir une topologie appropriée, vous devez tenir compte de vos exigences en matière de contrôle manuel, de taille du réseau, de fréquence de changement, de disponibilité et d'évolutivité.

### Avant de commencer

Cette tâche suppose que vous comprenez ce qu'est un réseau de publication / abonnement distribué et comment il fonctionne. Pour une présentation technique, voir [Réseaux de publication / abonnement distribués](#).

### Pourquoi et quand exécuter cette tâche

Il existe trois topologies de base pour un réseau de publication / abonnement:

- Cluster routé directement
- Cluster routé par l'hôte de rubrique
- Hiérarchie

Pour les deux premières topologies, le point de départ est une configuration de cluster IBM MQ . La troisième topologie peut être créée avec ou sans cluster. Pour plus d'informations sur la planification du réseau de gestionnaires de files d'attente sous-jacent, voir«[Planification de vos files d'attente réparties et de vos clusters](#)», à la page 21.

Un *cluster routé directement* est la topologie la plus simple à configurer lorsqu'un cluster est déjà présent. Toute rubrique que vous définissez sur un gestionnaire de files d'attente est automatiquement mise à disposition sur chaque gestionnaire de files d'attente du cluster, et les publications sont acheminées directement à partir de tout gestionnaire de files d'attente auquel une application de

publication se connecte, vers chacun des gestionnaires de files d'attente dans lesquels des abonnements correspondants existent. Cette simplicité de configuration repose sur le maintien par IBM MQ d'un niveau élevé de partage d'informations et de connectivité entre chaque gestionnaire de files d'attente du cluster. Pour les réseaux simples et de petite taille (c'est-à-dire un petit nombre de gestionnaires de files d'attente et un ensemble assez statique de diffuseurs et d'abonnés), cela est acceptable. Toutefois, lorsqu'il est utilisé dans des environnements plus grands ou plus dynamiques, le temps système peut être prohibitif. Voir [«Routage direct dans les clusters de publication / abonnement»](#), à la page 84.

Un *cluster routé par un hôte de rubrique* offre le même avantage qu'un cluster routé directement, en rendant disponible automatiquement sur chaque gestionnaire de files d'attente du cluster toute rubrique que vous définissez sur un gestionnaire de files d'attente du cluster. Toutefois, les clusters routés par l'hôte de rubrique nécessitent que vous choisissiez soigneusement les gestionnaires de files d'attente qui hébergent chaque rubrique, car toutes les informations et publications de cette rubrique transitent par ces gestionnaires de files d'attente hôte de rubrique. Cela signifie que le système n'a pas besoin de gérer les canaux et les flux d'informations entre tous les gestionnaires de files d'attente. Toutefois, cela signifie également que les publications peuvent ne plus être envoyées directement aux abonnés, mais qu'elles peuvent être acheminées via un gestionnaire de files d'attente hôte de rubrique. Pour ces raisons, une charge supplémentaire peut être placée sur le système, en particulier sur les gestionnaires de files d'attente hébergeant les rubriques. Par conséquent, une planification minutieuse de la topologie est requise. Cette topologie est particulièrement efficace pour les réseaux qui contiennent de nombreux gestionnaires de files d'attente ou qui hébergent un ensemble dynamique de diffuseurs et d'abonnés (c'est-à-dire des diffuseurs ou des abonnés fréquemment ajoutés ou supprimés). Des hôtes de rubrique supplémentaires peuvent être définis pour améliorer la disponibilité des routes et pour mettre à l'échelle horizontalement la charge de travail de publication. Voir [«Routage de l'hôte de rubrique dans les clusters de publication / abonnement»](#), à la page 89.

Une *hiérarchie* nécessite la configuration la plus manuelle à configurer et constitue la topologie la plus difficile à modifier. Vous devez configurer manuellement les relations entre chaque gestionnaire de files d'attente de la hiérarchie et ses relations directes. Une fois les relations configurées, les publications (comme pour les deux topologies précédentes) sont acheminées vers les abonnements des autres gestionnaires de files d'attente de la hiérarchie. Les publications sont acheminées à l'aide des relations de hiérarchie. Cela permet de configurer des topologies très spécifiques pour répondre à des exigences différentes, mais peut également entraîner des publications nécessitant de nombreux "tronçons" via des gestionnaires de files d'attente intermédiaires pour atteindre les abonnements. Il n'y a toujours qu'une seule route dans une hiérarchie pour une publication, de sorte que la disponibilité de chaque gestionnaire de files d'attente est critique. Les hiérarchies sont généralement préférables uniquement lorsqu'un seul cluster ne peut pas être configuré, par exemple lorsqu'il s'agit de plusieurs organisations. Voir [«Routage dans les hiérarchies de publication / abonnement»](#), à la page 115.

Si nécessaire, les trois topologies ci-dessus peuvent être combinées pour répondre à des besoins topographiques spécifiques. Pour un exemple, voir [Combinaison des espaces de sujet de plusieurs clusters](#).

Pour choisir une topologie adaptée à votre réseau de publication / abonnement distribué, vous devez prendre en compte les questions générales suivantes:

- Quelle sera la taille de votre réseau?
- De quel contrôle manuel avez-vous besoin sur sa configuration?
- Dans quelle mesure le système sera-t-il dynamique, tant en termes de rubriques et d'abonnements qu'en termes de gestionnaires de files d'attente?
- Quelles sont vos exigences en matière de disponibilité et d'évolutivité?
- Tous les gestionnaires de files d'attente peuvent-ils se connecter directement les uns aux autres?

## Procédure

- Estimez la taille de votre réseau.
  - a) Estimez le nombre de rubriques dont vous avez besoin.
  - b) Estimez le nombre de diffuseurs de publications et d'abonnés que vous prévoyez d'avoir.

c) Estimez le nombre de gestionnaires de files d'attente qui seront impliqués dans les activités de publication / abonnement.

Voir aussi [«Mise en cluster de publication / abonnement: meilleures pratiques»](#), à la page 99, en particulier les sections suivantes:

- [Comment dimensionner votre système](#)
- [Raisons de limiter le nombre de gestionnaires de files d'attente de cluster impliqués dans l'activité de publication / abonnement](#)
- [Comment choisir les rubriques à mettre en cluster](#)

Si votre réseau comporte de nombreux gestionnaires de files d'attente et gère de nombreux diffuseurs de publications et abonnés, vous devrez probablement utiliser un cluster ou une hiérarchie d'hôtes de rubriques. Les clusters routés directement ne nécessitent pratiquement aucune configuration manuelle et peuvent être une bonne solution pour les réseaux de petite taille ou statiques.

- Prenez en compte le contrôle manuel dont vous avez besoin sur le gestionnaire de files d'attente qui héberge chaque rubrique, diffuseur de publications ou abonné.
  - a) Déterminez si certains de vos gestionnaires de files d'attente sont moins capables que d'autres.
  - b) Déterminez si les liens de communication vers certains de vos gestionnaires de files d'attente sont plus fragiles que vers d'autres.
  - c) Identifiez les cas où vous vous attendez à ce qu'un sujet ait de nombreuses publications et peu d'abonnés.
  - d) Identifiez les cas où vous vous attendez à ce qu'un sujet ait de nombreux abonnés et peu de publications.

Dans toutes les topologies, les publications sont distribuées aux abonnements des autres gestionnaires de files d'attente. Dans un cluster routé directement, ces publications empruntent le chemin le plus court vers les abonnements. Dans un cluster routé par un hôte de rubrique ou une hiérarchie, vous contrôlez la route empruntée par les publications. Si vos gestionnaires de files d'attente ont des capacités différentes ou des niveaux de disponibilité et de connectivité différents, vous souhaiterez probablement affecter des charges de travail spécifiques à des gestionnaires de files d'attente spécifiques. Vous pouvez effectuer cette opération à l'aide d'un cluster routé par un hôte de rubrique ou d'une hiérarchie.

Dans toutes les topologies, la colocalisation des applications de publication sur le même gestionnaire de files d'attente que les abonnements permet de minimiser les surcharges et d'optimiser les performances. Pour les clusters routés par des hôtes de rubrique, pensez à placer des diffuseurs ou des abonnés sur les gestionnaires de files d'attente qui hébergent la rubrique. Cela supprime les "sauts" supplémentaires entre les gestionnaires de files d'attente pour transmettre une publication à un abonné. Cette approche est particulièrement efficace dans les cas où un sujet a beaucoup d'éditeurs et peu d'abonnés, ou beaucoup d'abonnés et peu d'éditeurs. Voir, par exemple, [Routage de l'hôte de rubrique à l'aide de diffuseurs de publications ou d'abonnés centralisés](#).

Voir aussi [«Mise en cluster de publication / abonnement: meilleures pratiques»](#), à la page 99, en particulier les sections suivantes:

- [Comment choisir les rubriques à mettre en cluster](#)
- [Emplacement du diffuseur de publications et de l'abonnement](#)
- Tenez compte de la dynamique de l'activité du réseau.
  - a) Estimez la fréquence à laquelle les abonnés seront ajoutés et supprimés sur différentes rubriques.

Chaque fois qu'un abonnement est ajouté ou supprimé d'un gestionnaire de files d'attente et qu'il s'agit du premier ou du dernier abonnement pour cette chaîne de rubrique spécifique, ces informations sont communiquées aux autres gestionnaires de files d'attente de la topologie. Dans un cluster routé directement et une hiérarchie, ces informations d'abonnement sont propagées à tous les gestionnaires de files d'attente de la topologie, qu'ils aient ou non des diffuseurs sur la rubrique. Si la topologie comprend de nombreux gestionnaires de files d'attente, cela peut entraîner une surcharge de performances importante. Dans un cluster routé par un hôte de



rubrique, ces informations sont uniquement propagées aux gestionnaires de files d'attente qui hébergent une rubrique en cluster mappée à la chaîne de rubrique de l'abonnement.

Voir aussi la section [Changement d'abonnement et chaînes de rubrique dynamiques de « Mise en cluster de publication / abonnement: meilleures pratiques »](#), à la page 99.

**Remarque :** Dans les systèmes très dynamiques, où l'ensemble de nombreuses chaînes de rubrique uniques est rapidement et constamment modifié, il peut être préférable de basculer le modèle vers un mode de "publication partout". Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

b) Tenez compte de la dynamique des gestionnaires de files d'attente dans la topologie.

Une hiérarchie requiert que chaque modification apportée au gestionnaire de files d'attente dans la topologie soit insérée manuellement ou supprimée de la hiérarchie, avec précaution lors de la modification des gestionnaires de files d'attente à des niveaux supérieurs de la hiérarchie. Les gestionnaires de files d'attente d'une hiérarchie utilisent généralement également des connexions de canal configurées manuellement. Vous devez gérer ces connexions, en ajoutant et en supprimant des canaux lorsque des gestionnaires de files d'attente sont ajoutés et supprimés de la hiérarchie.

Dans un cluster de publication / abonnement, les gestionnaires de files d'attente sont automatiquement connectés à tout autre gestionnaire de files d'attente requis lorsqu'ils rejoignent le cluster pour la première fois et prennent automatiquement en compte les rubriques et les abonnements.

- Tenez compte de la disponibilité de votre route et des exigences d'évolutivité du trafic de publication.
  - a) Déterminez si vous devez toujours disposer d'une route disponible entre un gestionnaire de files d'attente de publication et un gestionnaire de files d'attente d'abonnement, même si un gestionnaire de files d'attente n'est pas disponible.
  - b) Réfléchissez à la mesure dans laquelle vous avez besoin que le réseau soit évolutif. Déterminez si le niveau de trafic de publication est trop élevé pour être acheminé via un seul gestionnaire de files d'attente ou canal et si ce niveau de trafic de publication doit être géré par une seule branche de rubrique ou s'il peut être réparti entre plusieurs branches de rubrique.
  - c) Déterminez si vous devez conserver l'ordre des messages.

Etant donné qu'un cluster routé directement envoie des messages directement des gestionnaires de files d'attente de publication aux gestionnaires de files d'attente d'abonnement, vous n'avez pas besoin de prendre en compte la disponibilité des gestionnaires de files d'attente intermédiaires le long de la route. De même, la mise à l'échelle vers les gestionnaires de files d'attente intermédiaires n'est pas prise en compte. Toutefois, comme indiqué précédemment, la surcharge liée à la gestion automatique des canaux et des flux d'informations entre tous les gestionnaires de files d'attente du cluster peut affecter de manière significative les performances, en particulier dans un environnement dynamique ou de grande taille.

Un cluster routé par un hôte de rubrique peut être optimisé pour des rubriques individuelles. Vous pouvez vous assurer que chaque branche de l'arborescence de rubriques ayant une charge de travail de publication considérable est définie sur un gestionnaire de files d'attente différent et que chaque gestionnaire de files d'attente est suffisamment performant et disponible pour la charge de travail attendue pour cette branche de l'arborescence de rubriques. Vous pouvez également améliorer la disponibilité et la mise à l'échelle horizontale en définissant chaque rubrique sur plusieurs gestionnaires de files d'attente. Cela permet au système de router les gestionnaires de files d'attente d'hôte de rubrique non disponibles et d'équilibrer la charge de travail du trafic de publication entre ces gestionnaires. Toutefois, lorsque vous définissez une rubrique donnée sur plusieurs gestionnaires de files d'attente, vous introduisez également les contraintes suivantes:

- Vous perdez l'ordre des messages dans les publications.
- Vous ne pouvez pas utiliser les publications conservées. Voir [« Remarques de conception pour les publications conservées dans les clusters de publication/abonnement »](#), à la page 113.

Vous ne pouvez pas configurer la haute disponibilité ou l'évolutivité du routage dans une hiérarchie via plusieurs routes.

Voir aussi la section [Trafic de publication de «Mise en cluster de publication / abonnement: meilleures pratiques»](#), à la page 99.

- En fonction de ces calculs, utilisez les liens fournis pour vous aider à décider si vous souhaitez utiliser un cluster routé par un hôte de rubrique, un cluster routé directement, une hiérarchie ou un mélange de ces topologies.

## Que faire ensuite

Vous êtes maintenant prêt à configurer votre réseau de publication / abonnement distribué.

### Tâches associées

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

[Configuration de la file d'attente répartie](#)

[Configuration d'un cluster de publication / abonnement](#)

[Connexion d'un gestionnaire de files d'attente à une hiérarchie de publication/abonnement](#)

## Conception des clusters de publication/abonnement

Il existe deux topologies de cluster de publication / abonnement de base: *routage direct* et *routage via un hôte de rubrique*. Chacun a des avantages différents. Lorsque vous concevez votre cluster de publication / abonnement, choisissez la topologie qui correspond le mieux aux exigences de votre réseau.

Pour une présentation des deux topologies de cluster de publication / abonnement, voir [Clusters de publication / abonnement](#). Pour vous aider à évaluer les exigences de votre réseau, voir [«Planification de votre réseau de publication / abonnement distribué»](#), à la page 78 et [«Mise en cluster de publication / abonnement: meilleures pratiques»](#), à la page 99.

En général, les deux topologies de cluster offrent les avantages suivants:

- Configuration simple sur une topologie de cluster point à point.
- Traitement automatique des gestionnaires de files d'attente qui rejoignent et quittent le cluster.
- Facilité de mise à l'échelle pour des abonnements et des diffuseurs supplémentaires, en ajoutant des gestionnaires de files d'attente supplémentaires et en répartissant les abonnements et diffuseurs supplémentaires entre eux.

Cependant, les deux topologies présentent des avantages différents au fur et à mesure que les exigences deviennent plus spécifiques.

## Clusters de publication / abonnement routés directement

Avec le routage direct, tout gestionnaire de files d'attente du cluster envoie les publications des applications connectées directement à tout autre gestionnaire de files d'attente du cluster avec un abonnement correspondant.

Un cluster de publication / abonnement routé directement offre les avantages suivants:

- Les messages destinés à un abonnement sur un gestionnaire de files d'attente spécifique dans le même cluster sont transportés directement vers ce gestionnaire de files d'attente et n'ont pas besoin de passer par un gestionnaire de files d'attente intermédiaire. Cela peut améliorer les performances par rapport à une topologie routée par un hôte de rubrique ou à une topologie hiérarchique.
- Étant donné que tous les gestionnaires de files d'attente sont directement connectés les uns aux autres, il n'existe pas de point de défaillance unique dans l'infrastructure de routage de cette topologie. Si un gestionnaire de files d'attente n'est pas disponible, les abonnements des autres gestionnaires de files d'attente du cluster peuvent toujours recevoir des messages des diffuseurs de publications des gestionnaires de files d'attente disponibles.
- Il est très simple à configurer, en particulier sur un cluster existant.

Éléments à prendre en compte lors de l'utilisation d'un cluster de publication / abonnement routé directement:

- Tous les gestionnaires de files d'attente du cluster prennent connaissance de tous les autres gestionnaires de files d'attente du cluster.
- Les gestionnaires de files d'attente d'un cluster qui hébergent un ou plusieurs abonnements à une rubrique en cluster créent automatiquement des canaux émetteurs de cluster pour tous les autres gestionnaires de files d'attente du cluster, même lorsque ces gestionnaires de files d'attente ne publient pas de messages sur des rubriques en cluster.
- Le premier abonnement d'un gestionnaire de files d'attente à une chaîne de rubrique sous une rubrique en cluster entraîne l'envoi d'un message à tous les autres gestionnaires de files d'attente du cluster. De même, le dernier abonnement sur une chaîne de rubrique à supprimer génère également un message. Plus les chaînes de rubrique individuelles sont utilisées sous une rubrique en cluster et plus le taux de modification des abonnements est élevé, plus la communication entre les gestionnaires de files d'attente est importante.
- Chaque gestionnaire de files d'attente du cluster conserve la connaissance des chaînes de rubrique souscrites dont il est informé, même lorsque le gestionnaire de files d'attente ne publie ni ne s'abonne à ces rubriques.

Pour les raisons ci-dessus, tous les gestionnaires de files d'attente d'un cluster avec une rubrique de routage direct définie entraînent une surcharge supplémentaire. Plus il y a de gestionnaires de files d'attente dans le cluster, plus le temps système est élevé. De même, plus le nombre de chaînes de rubrique souscrites est élevé et plus le taux de modification est élevé, plus le temps système est élevé. Cela peut entraîner une charge trop importante sur les gestionnaires de files d'attente s'exécutant sur de petits systèmes dans un cluster de publication / abonnement direct routé de grande taille ou dynamique. Pour plus d'informations, voir [Direct route publish / subscribe performance](#).

Lorsque vous savez qu'un cluster ne peut pas prendre en charge les surcharges de la publication / abonnement en cluster routé directement, vous pouvez utiliser à la place [l'hôte de rubrique routé par la publication / abonnement](#). Dans des situations extrêmes, vous pouvez également désactiver complètement la fonctionnalité de publication / abonnement en cluster en définissant l'attribut de gestionnaire de files d'attente **PSCLUS** sur DISABLED sur chaque gestionnaire de files d'attente du cluster. Voir «[Blocage de la publication / l'abonnement en cluster](#)», à la page 110. Cela permet d'éviter la création d'une rubrique en cluster et de s'assurer que votre réseau n'entraîne pas de surcharge associée à la publication / l'abonnement en cluster.

## Clusters de publication / abonnement routés par l'hôte de rubrique

Avec le routage via un hôte de rubrique, les gestionnaires de files d'attente dans lesquels des rubriques en cluster sont définies de manière administrative deviennent des routeurs pour les publications. Les publications provenant de gestionnaires de files d'attente non hébergeant dans le cluster sont acheminées via le gestionnaire de files d'attente hôte vers n'importe quel gestionnaire de files d'attente du cluster avec un abonnement correspondant.

Un cluster de publication / abonnement routé par un hôte de rubrique offre les avantages supplémentaires suivants par rapport à un cluster de publication / abonnement routé directement:

- Seuls les gestionnaires de files d'attente sur lesquels sont définies les rubriques routées par l'hôte de rubriques sont informés de tous les autres gestionnaires de files d'attente du cluster.
- Seuls les gestionnaires de files d'attente hôte de rubrique doivent pouvoir se connecter à tous les autres gestionnaires de files d'attente du cluster et se connectent généralement uniquement à ceux où des abonnements existent. Par conséquent, il y a beaucoup moins de canaux en cours d'exécution entre les gestionnaires de files d'attente.
- Les gestionnaires de files d'attente de cluster qui hébergent un ou plusieurs abonnements à une rubrique de cluster créent automatiquement des canaux émetteurs de cluster uniquement aux gestionnaires de files d'attente qui hébergent une rubrique de cluster mappée à la chaîne de rubrique de l'abonnement.
- Le premier abonnement d'un gestionnaire de files d'attente à une chaîne de rubrique sous une rubrique en cluster entraîne l'envoi d'un message à un gestionnaire de files d'attente dans le cluster qui héberge la rubrique en cluster. De même, le dernier abonnement sur une chaîne de rubrique à supprimer génère également un message. Plus les chaînes de rubrique individuelles sont utilisées sous une rubrique en

cluster et plus le taux de changement des abonnements est élevé, plus la communication entre les gestionnaires de files d'attente est importante, mais uniquement entre les hôtes d'abonnement et les hôtes de rubrique.

- Contrôle accru sur la configuration physique. Avec le routage direct, tous les gestionnaires de files d'attente doivent participer au cluster de publication / abonnement, ce qui augmente leurs surcharges. Avec le routage via un hôte de rubrique, seuls les gestionnaires de files d'attente hôte de rubrique connaissent les autres gestionnaires de files d'attente et leurs abonnements. Vous choisissez explicitement les gestionnaires de files d'attente hôte de rubrique ; par conséquent, vous pouvez vous assurer que ces gestionnaires de files d'attente s'exécutent sur un équipement adéquat et vous pouvez utiliser des systèmes moins puissants pour les autres gestionnaires de files d'attente.

Éléments à prendre en compte lors de l'utilisation d'un cluster de publication / abonnement routé par un hôte de rubrique:

- Un "tronçon" supplémentaire entre un gestionnaire de files d'attente de publication et un gestionnaire de files d'attente d'abonnement est introduit lorsque le diffuseur de publications ou l'abonné ne se trouve pas sur un gestionnaire de files d'attente d'hébergement de rubriques. Le temps d'attente provoqué par le "tronçon" supplémentaire peut signifier que le routage de l'hôte de rubrique est moins efficace que le routage direct.
- Sur les clusters de grande taille, le routage par hôte de rubrique facilite les problèmes de performances et de mise à l'échelle importants que vous pouvez obtenir avec le routage direct.
- Vous pouvez choisir de définir toutes vos rubriques sur un seul gestionnaire de files d'attente ou sur un très petit nombre de gestionnaires de files d'attente. Dans ce cas, assurez-vous que les gestionnaires de files d'attente hôte de rubrique sont hébergés sur des systèmes puissants disposant d'une bonne connectivité.
- Vous pouvez définir la même rubrique sur plusieurs gestionnaires de files d'attente. Cela améliore la disponibilité de la rubrique ainsi que l'évolutivité car la charge de travail IBM MQ équilibre les publications d'une rubrique sur tous les hôtes de cette rubrique. Notez toutefois que la définition de la même rubrique sur plusieurs gestionnaires de files d'attente perd l'ordre des messages pour cette rubrique.
- En hébergeant différentes rubriques sur différents gestionnaires de files d'attente, vous pouvez améliorer l'évolutivité sans perdre l'ordre des messages.

### **Tâches associées**

[Scénario: Création d'un cluster de publication / abonnement](#)

[Configuration d'un cluster de publication / abonnement](#)

[Optimisation des réseaux de publication / abonnement distribués](#)

[Identification et résolution des problèmes de publication / abonnement distribué](#)

### ***Routage direct dans les clusters de publication / abonnement***

Les publications provenant de n'importe quel gestionnaire de files d'attente de publication sont acheminées directement vers n'importe quel autre gestionnaire de files d'attente du cluster avec un abonnement correspondant.

Pour une présentation de la façon dont les messages sont acheminés entre les gestionnaires de files d'attente dans les hiérarchies de publication / abonnement et les clusters, voir [Réseaux de publication / abonnement distribués](#).

Un cluster de publication / abonnement routé directement se comporte comme suit:

- Tous les gestionnaires de files d'attente connaissent automatiquement tous les autres gestionnaires de files d'attente.
- Tous les gestionnaires de files d'attente avec des abonnements à des rubriques en cluster créent des canaux pour tous les autres gestionnaires de files d'attente du cluster et les informent de leurs abonnements.

- Les messages publiés par une application sont acheminés depuis le gestionnaire de files d'attente auquel elle est connectée, directement vers chaque gestionnaire de files d'attente dans lequel existe un abonnement correspondant.

Le diagramme suivant illustre un cluster de gestionnaires de files d'attente qui n'est pas actuellement utilisé pour des activités de publication / abonnement ou de point à point. Notez que chaque gestionnaire de files d'attente du cluster se connecte uniquement aux gestionnaires de files d'attente de référentiel complet et à partir de ces derniers.

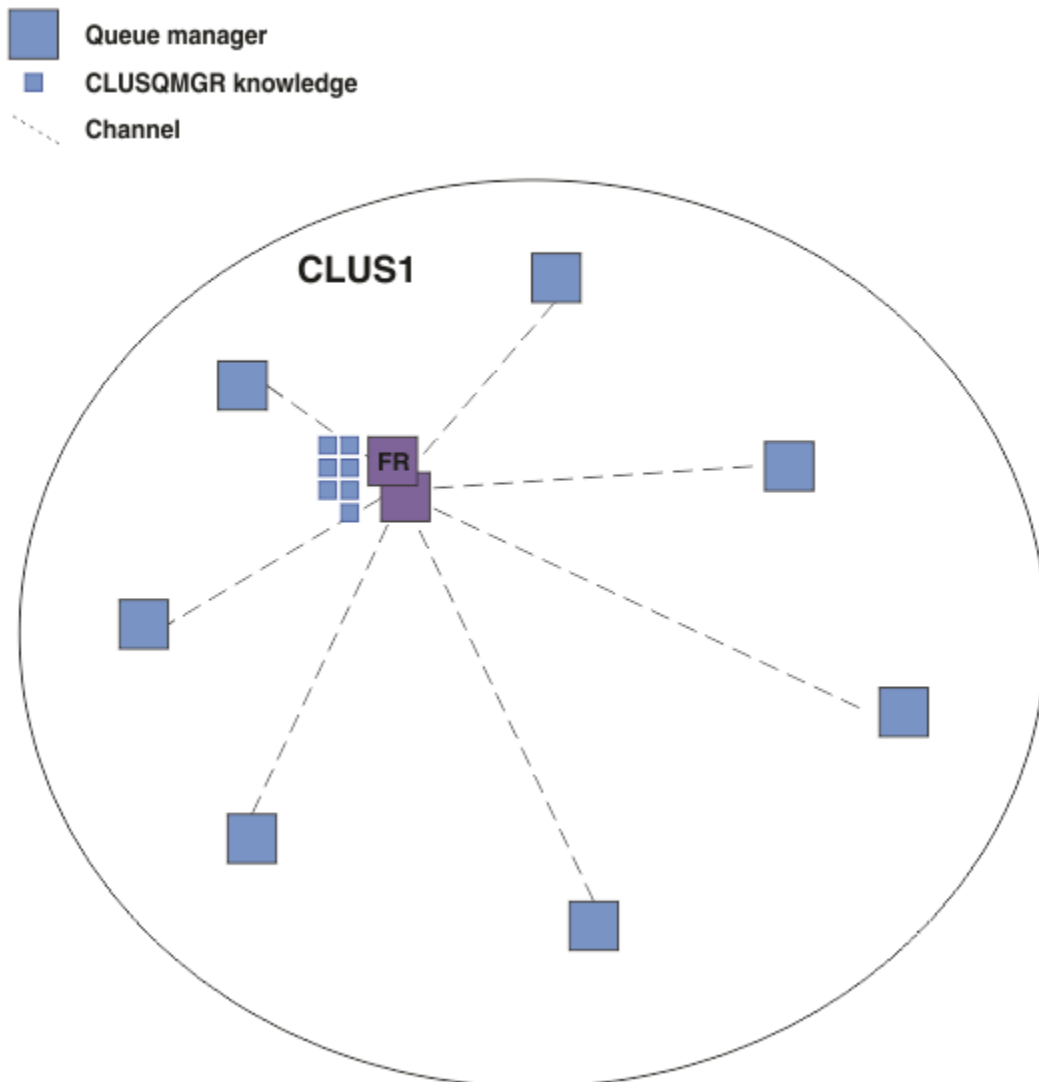


Figure 16. Un cluster de gestionnaires de files d'attente

Pour que les publications circulent entre les gestionnaires de files d'attente dans un cluster routé directement, vous devez mettre en cluster une branche de l'arborescence de rubriques, comme décrit dans la rubrique [Configuration d'un cluster de publication / abonnement](#), et spécifier le *routage direct* (valeur par défaut).

Dans un cluster de publication / abonnement routé directement, vous définissez l'objet de rubrique sur n'importe quel gestionnaire de files d'attente du cluster. Lorsque vous effectuez cette opération, la connaissance de l'objet et de tous les autres gestionnaires de files d'attente du cluster est automatiquement envoyée à tous les gestionnaires de files d'attente du cluster par les gestionnaires de files d'attente de référentiel complet. Cela se produit avant qu'un gestionnaire de files d'attente ne fasse référence à la rubrique:

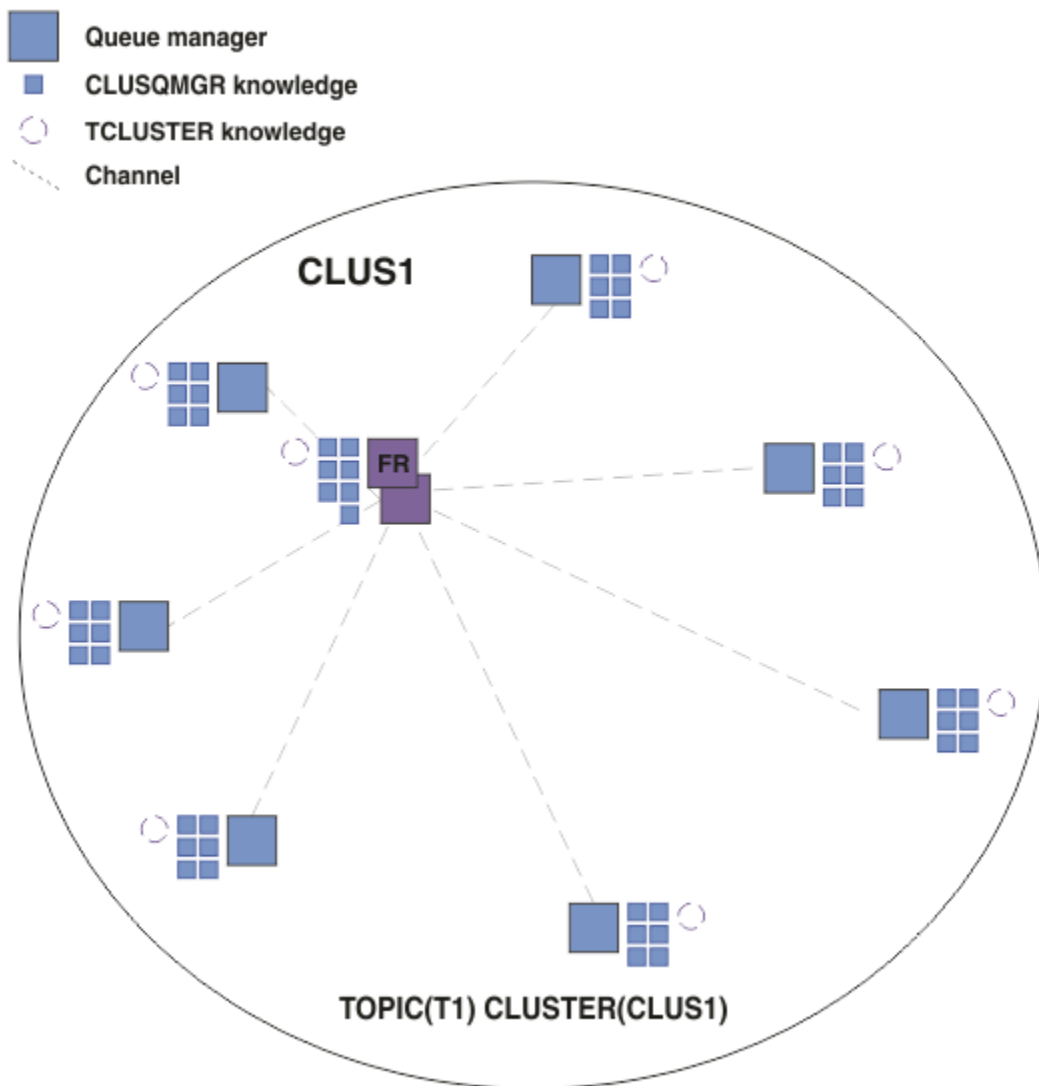


Figure 17. Un cluster de publication / abonnement routé directement

Lorsqu'un abonnement est créé, le gestionnaire de files d'attente qui héberge l'abonnement établit un canal pour chaque gestionnaire de files d'attente du cluster et envoie les détails de l'abonnement. Cette connaissance des abonnements distribués est représentée par un abonnement de proxy sur chaque gestionnaire de files d'attente. Lorsqu'une publication est produite sur un gestionnaire de files d'attente du cluster qui correspond à la chaîne de rubrique de l'abonnement de proxy, un canal de cluster est établi entre le gestionnaire de files d'attente du diffuseur de publications et chaque gestionnaire de files d'attente hébergeant un abonnement et le message est envoyé à chacun d'eux.

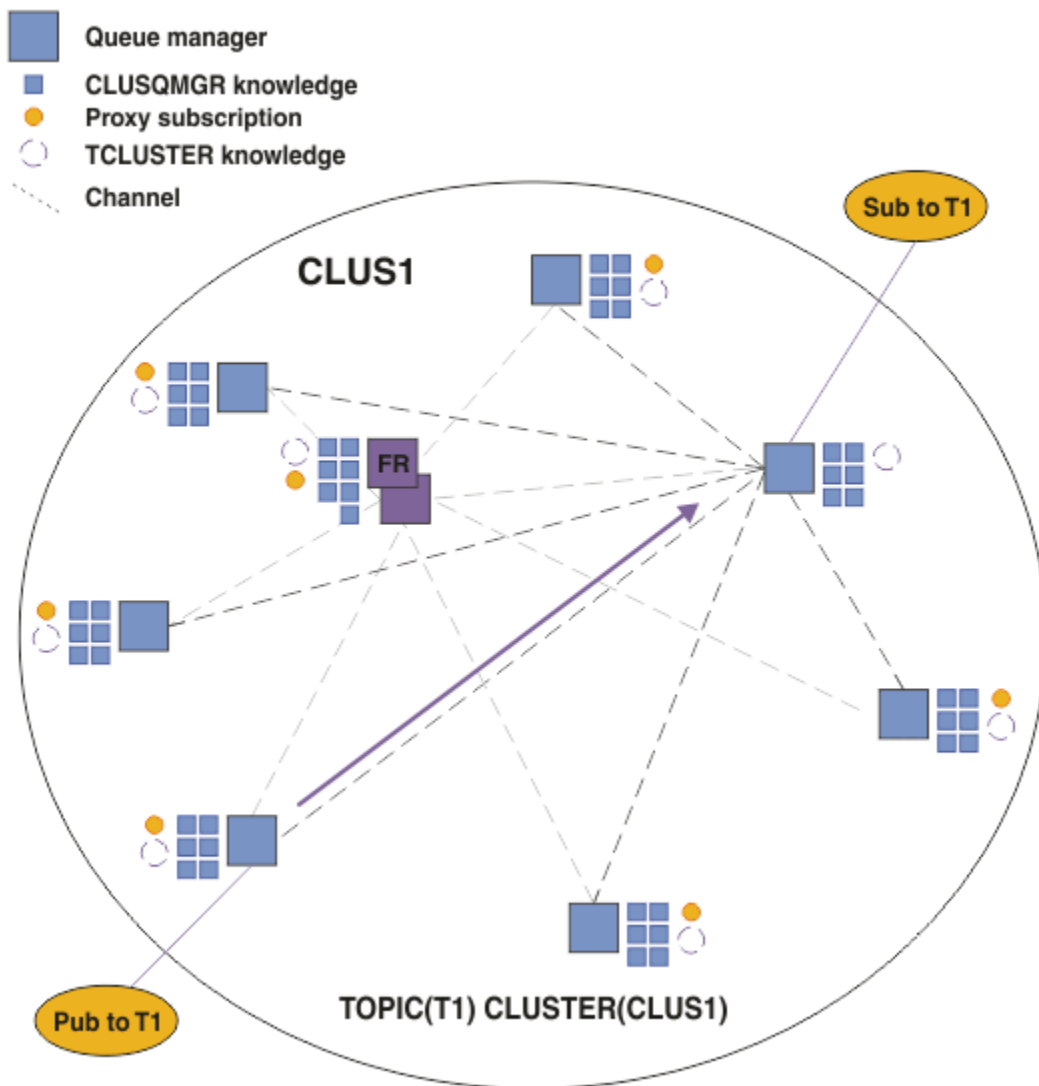


Figure 18. Un cluster de publication / abonnement routé directement avec un diffuseur de publications et un abonné à une rubrique en cluster

Le routage direct des publications vers les gestionnaires de files d'attente d'hébergement d'abonnement simplifie la configuration et réduit le temps d'attente lors de la distribution des publications aux abonnements.

Toutefois, en fonction de l'emplacement des abonnements et des diffuseurs, votre cluster peut rapidement devenir entièrement interconnecté, chaque gestionnaire de files d'attente ayant une connexion directe à tous les autres gestionnaires de files d'attente. Cela peut être acceptable ou non dans votre environnement. De même, si l'ensemble des chaînes de rubrique auxquelles vous êtes abonné change fréquemment, la charge de propagation de ces informations entre tous les gestionnaires de files d'attente peut également devenir importante. Tous les gestionnaires de files d'attente d'un cluster de publication / abonnement routé directement doivent pouvoir faire face à ces surcharges.

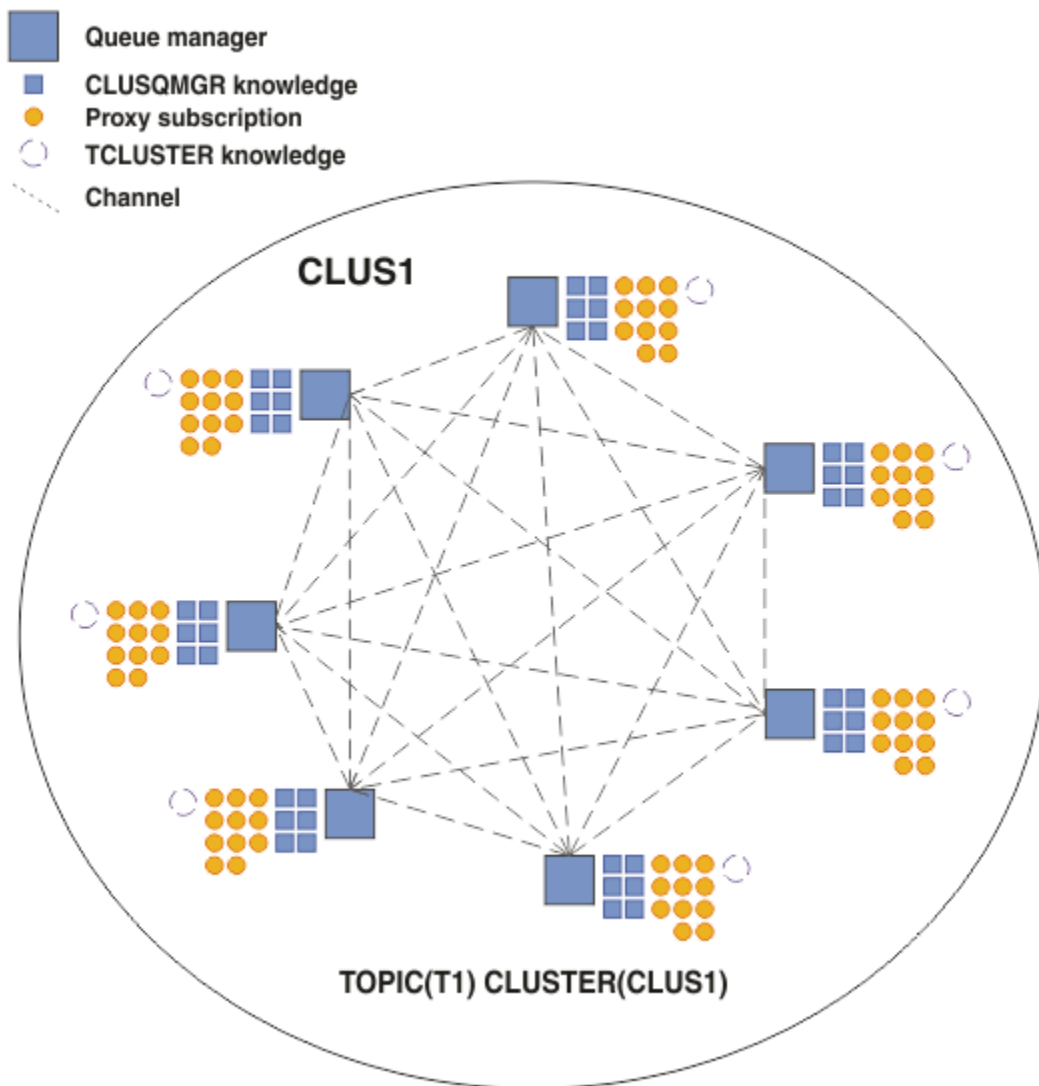


Figure 19. Un cluster de publication / abonnement routé directement qui est entièrement interconnecté

## Résumé et remarques supplémentaires

Un cluster de publication / abonnement routé directement nécessite peu d'intervention manuelle pour créer ou administrer, et fournit un routage direct entre les diffuseurs et les abonnés. Pour certaines configurations, il s'agit généralement de la topologie la plus appropriée, notamment les clusters avec peu de gestionnaires de files d'attente, ou pour lesquels une connectivité de gestionnaire de files d'attente élevée est acceptable et les abonnements changent rarement. Cependant, il impose également certaines contraintes à votre système:

- La charge sur chaque gestionnaire de files d'attente est proportionnelle au nombre total de gestionnaires de files d'attente dans le cluster. Par conséquent, dans les grands clusters, les gestionnaires de files d'attente individuels et le système dans son ensemble peuvent rencontrer des problèmes de performances.
- Par défaut, toutes les chaînes de rubrique en cluster auxquelles vous êtes abonné sont propagées dans le cluster, et les publications sont propagées uniquement aux gestionnaires de files d'attente éloignées qui ont un abonnement à la rubrique associée. Par conséquent, des changements rapides de l'ensemble des abonnements peuvent devenir un facteur limitatif. Vous pouvez modifier ce comportement par défaut et faire en sorte que toutes les publications soient propagées à tous les gestionnaires de files d'attente, ce qui élimine le besoin d'abonnements de proxy. 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](#).

**Remarque :** Une restriction similaire s'applique également aux hiérarchies.

- En raison de la nature interconnectée des gestionnaires de files d'attente de publication / abonnement, la propagation des abonnements proxy sur tous les noeuds du réseau prend du temps. Les publications distantes ne commencent pas nécessairement à être abonnées immédiatement, de sorte que les publications précoces peuvent ne pas être envoyées à la suite d'un abonnement à une nouvelle chaîne de rubrique. Vous pouvez supprimer les problèmes causés par le délai d'abonnement en propageant toutes les publications à tous les gestionnaires de files d'attente, ce qui élimine le besoin d'abonnements de proxy. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

**Remarque :** Cette restriction s'applique également aux hiérarchies.

Avant d'utiliser le routage direct, explorez les approches alternatives détaillées dans «[Routage de l'hôte de rubrique dans les clusters de publication / abonnement](#)», à la page 89 et «[Routage dans les hiérarchies de publication / abonnement](#)», à la page 115.

### ***Routage de l'hôte de rubrique dans les clusters de publication / abonnement***

Les publications provenant de gestionnaires de files d'attente non hébergeant dans le cluster sont acheminées via le gestionnaire de files d'attente hôte vers n'importe quel gestionnaire de files d'attente du cluster avec un abonnement correspondant.

Pour une présentation de la façon dont les messages sont acheminés entre les gestionnaires de files d'attente dans les hiérarchies de publication / abonnement et les clusters, voir [Réseaux de publication / abonnement distribués](#).

Pour comprendre le comportement et les avantages du routage via un hôte de rubrique, il est préférable de comprendre «[Routage direct dans les clusters de publication / abonnement](#)», à la page 84.

Un cluster de publication / abonnement routé par un hôte de rubrique se comporte comme suit:

- Les objets de rubrique gérés en cluster sont définis manuellement sur des gestionnaires de files d'attente individuels dans le cluster. Ils sont appelés *gestionnaires de files d'attente hôte de rubrique*.
- Lorsqu'un abonnement est effectué sur un gestionnaire de files d'attente de cluster, les canaux sont créés depuis le gestionnaire de files d'attente hôte d'abonnement vers les gestionnaires de files d'attente hôte de rubrique, et les abonnements de proxy sont créés uniquement sur les gestionnaires de files d'attente qui hébergent la rubrique.
- Lorsqu'une application publie des informations dans une rubrique, le gestionnaire de files d'attente connecté réachemine toujours la publication vers un gestionnaire de files d'attente qui héberge la rubrique, qui la transmet à tous les gestionnaires de files d'attente du cluster ayant des abonnements correspondants à la rubrique.

Ce processus est expliqué plus en détail dans les exemples suivants.

### **Routage de l'hôte de rubrique à l'aide d'un hôte de rubrique unique**

Pour que les publications circulent entre les gestionnaires de files d'attente d'un cluster routé par un hôte de rubrique, vous devez mettre en cluster une branche de l'arborescence de rubriques, comme décrit dans la rubrique [Configuration d'un cluster de publication / abonnement](#), et spécifier le *routage via un hôte de rubrique*.

Il existe un certain nombre de raisons de définir un objet de rubrique routé par un hôte de rubrique sur plusieurs gestionnaires de files d'attente dans un cluster. Toutefois, par souci de simplicité, nous commençons par un hôte de rubrique unique.

Le diagramme suivant illustre un cluster de gestionnaires de files d'attente qui n'est pas actuellement utilisé pour des activités de publication / abonnement ou de point à point. Notez que chaque gestionnaire de files d'attente du cluster se connecte uniquement aux gestionnaires de files d'attente de référentiel complet et à partir de ces derniers.

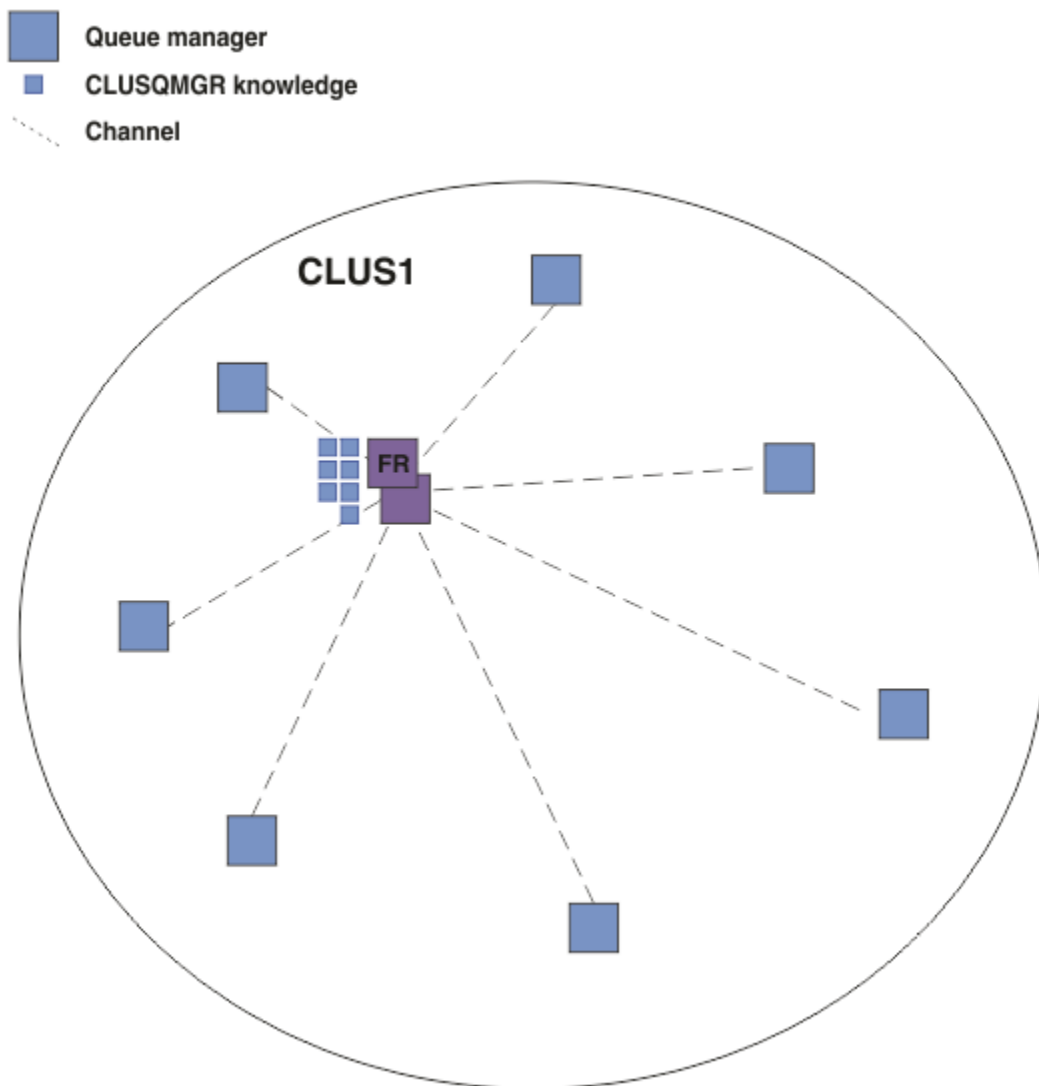


Figure 20. Un cluster de gestionnaires de files d'attente

Dans un cluster de publication / abonnement routé par un hôte de rubrique, vous définissez l'objet de rubrique sur un gestionnaire de files d'attente spécifique dans le cluster. Le trafic de publication / abonnement passe ensuite par ce gestionnaire de files d'attente, ce qui en fait un gestionnaire de files d'attente critique dans le cluster et augmente sa charge de travail. Pour ces raisons, il n'est pas recommandé d'utiliser un gestionnaire de files d'attente de référentiel complet, mais d'utiliser un autre gestionnaire de files d'attente dans le cluster. Lorsque vous définissez l'objet de rubrique sur le gestionnaire de files d'attente hôte, la connaissance de l'objet et de son hôte est automatiquement envoyée par les gestionnaires de files d'attente de référentiel complet à tous les autres gestionnaires de files d'attente du cluster. A la différence du *routage direct*, chaque gestionnaire de files d'attente n'est pas informé des autres gestionnaires de files d'attente du cluster.

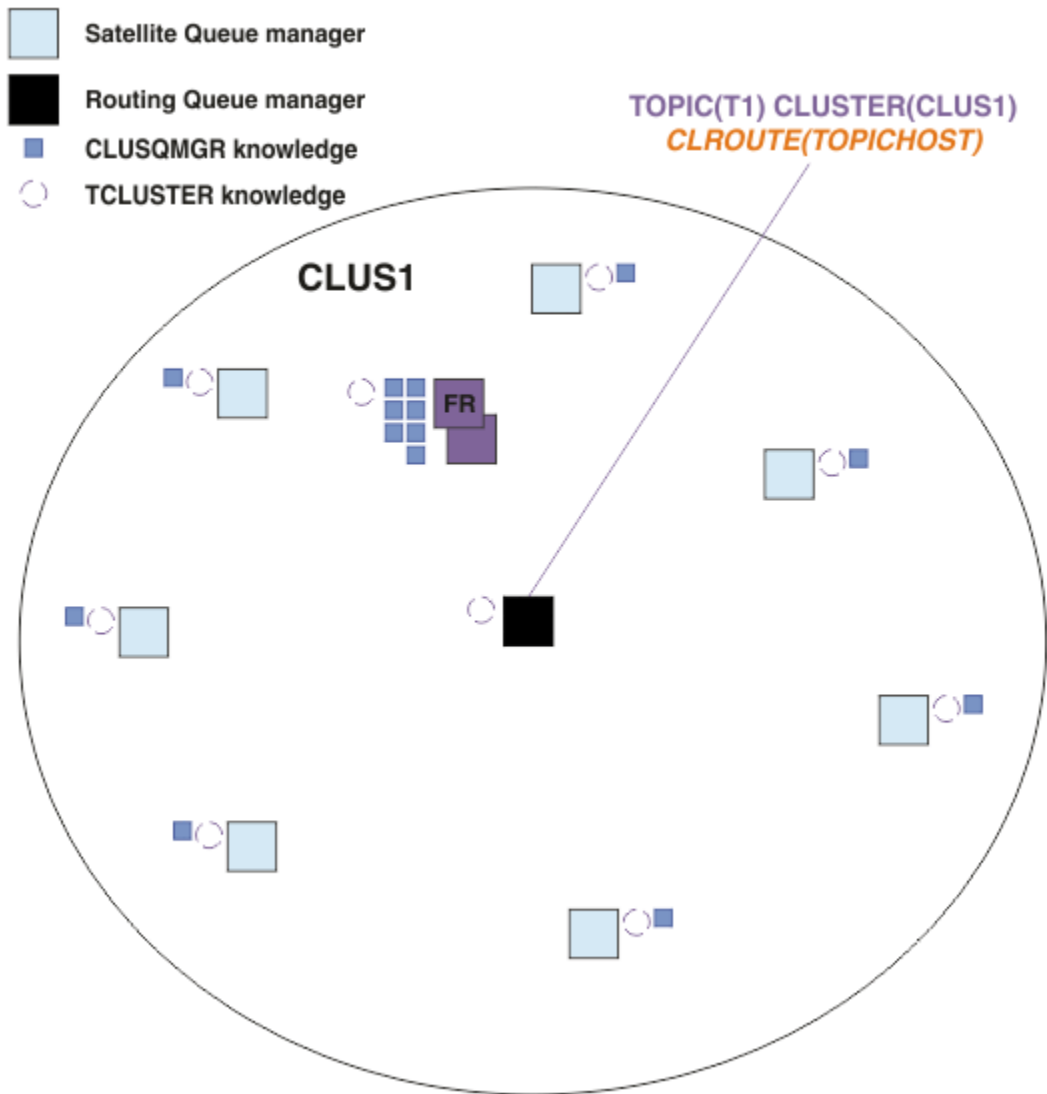


Figure 21. Un hôte de rubrique a routé un cluster de publication / abonnement avec une rubrique définie sur un hôte de rubrique

Lorsqu'un abonnement est créé sur un gestionnaire de files d'attente, un canal est créé entre le gestionnaire de files d'attente abonné et le gestionnaire de files d'attente hôte de rubrique. Le gestionnaire de files d'attente d'abonnement se connecte uniquement au gestionnaire de files d'attente hôte de rubrique et envoie les détails de l'abonnement (sous la forme d'un *abonnement de proxy*). Le gestionnaire de files d'attente hôte de rubrique ne transmet pas ces informations d'abonnement à d'autres gestionnaires de files d'attente du cluster.

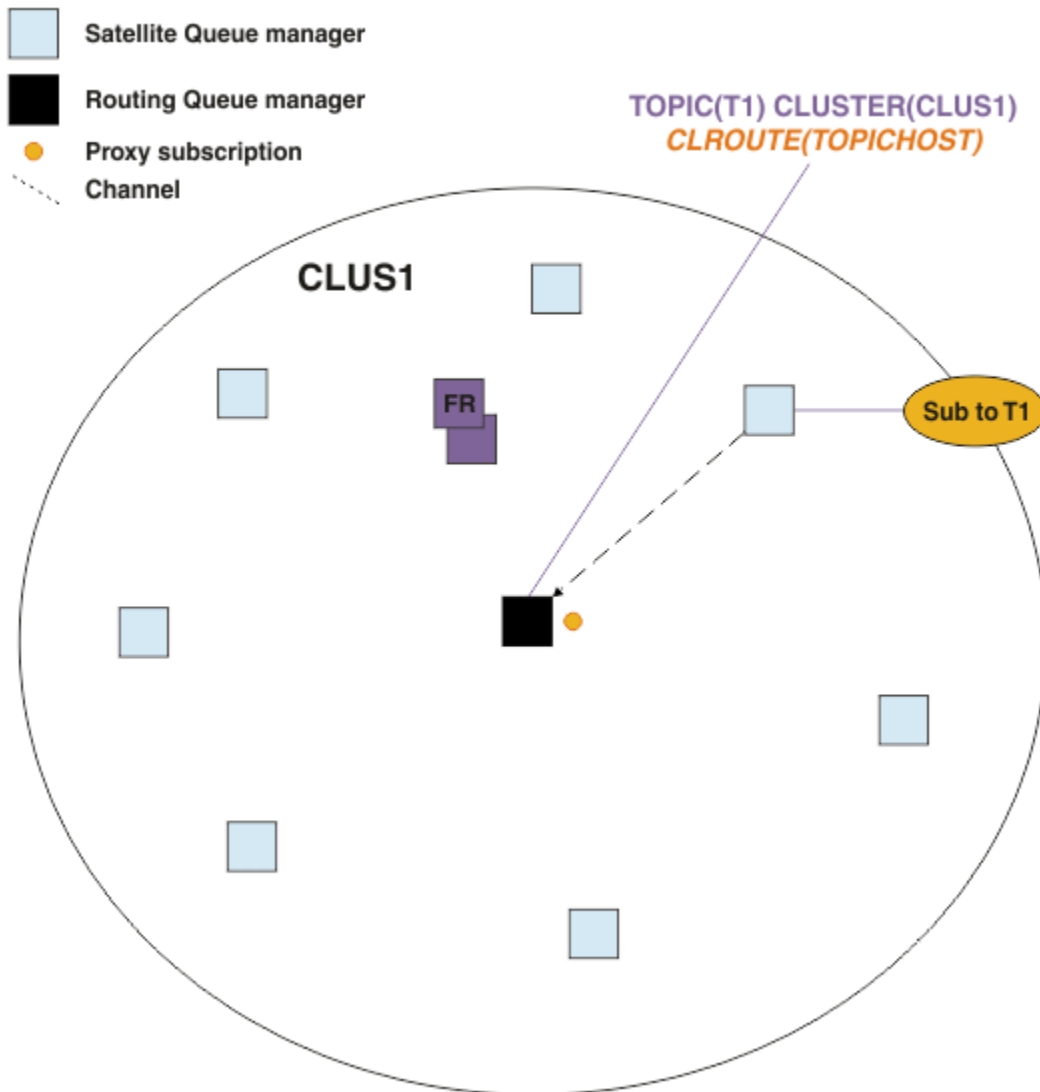


Figure 22. Un hôte de rubrique a routé un cluster de publication / abonnement avec une rubrique définie sur un hôte de rubrique et un abonné

Lorsqu'une application de publication se connecte à un autre gestionnaire de files d'attente et qu'un message est publié, un canal est créé entre le gestionnaire de files d'attente de publication et le gestionnaire de files d'attente hôte de rubrique, et le message est réacheminé vers ce gestionnaire de files d'attente. Le gestionnaire de files d'attente de publication n'ayant connaissance d'aucun abonnement sur d'autres gestionnaires de files d'attente du cluster, le message est réacheminé vers le gestionnaire de files d'attente hôte de rubrique même s'il n'y a pas d'abonnés à cette rubrique dans le cluster. Le gestionnaire de files d'attente de publication se connecte uniquement au gestionnaire de files d'attente hôte de rubrique. Les publications sont acheminées via l'hôte de rubrique vers les gestionnaires de files d'attente abonnés, le cas échéant.

Les abonnements sur le même gestionnaire de files d'attente que le diffuseur de publications sont satisfaits directement, sans envoyer d'abord les messages à un gestionnaire de files d'attente hôte de rubrique.

Notez que, en raison du rôle critique joué par chaque gestionnaire de files d'attente hôte de rubrique, vous devez choisir des gestionnaires de files d'attente capables de gérer les exigences de charge, de disponibilité et de connectivité de l'hébergement de rubrique.

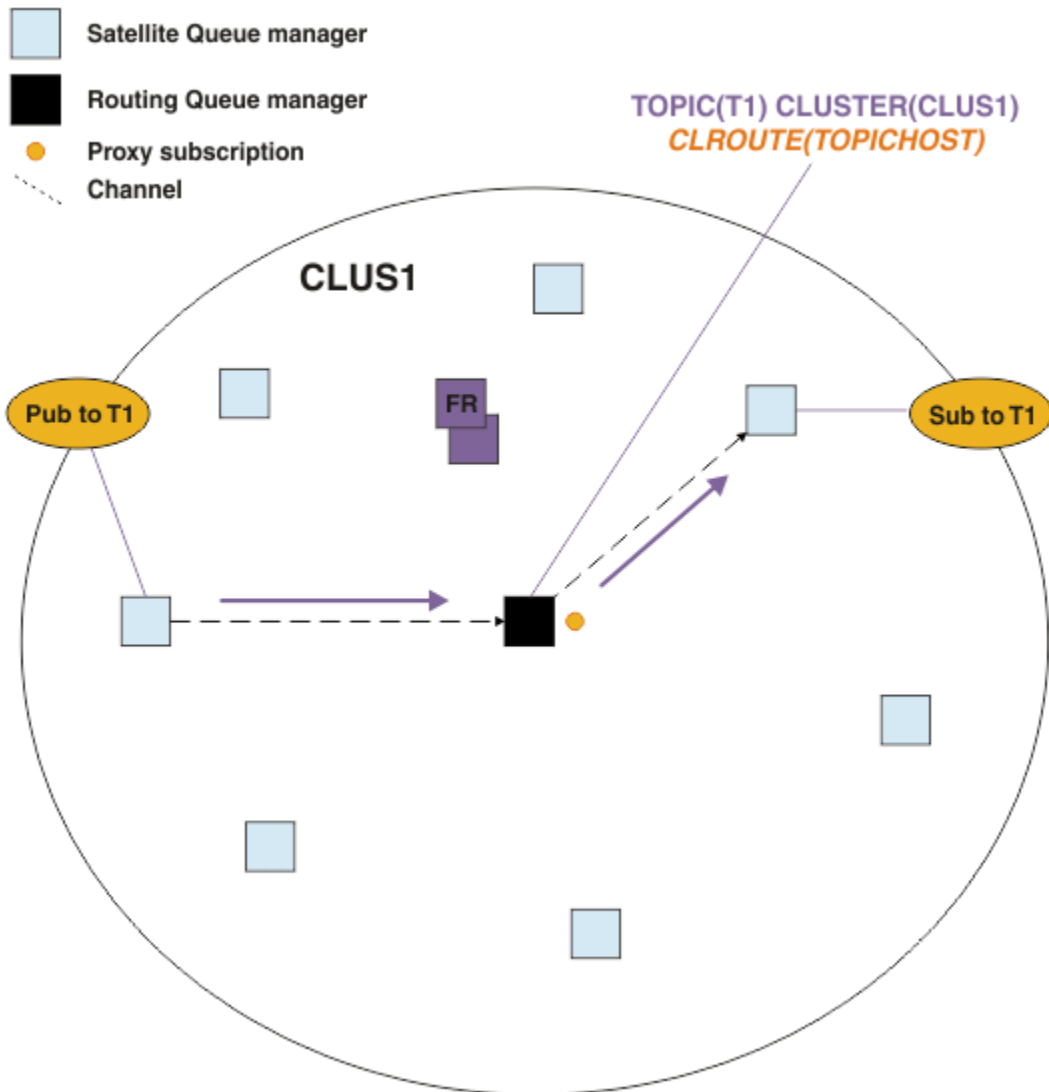


Figure 23. Un hôte de rubrique a routé un cluster de publication / abonnement avec une rubrique, un abonné et un diffuseur de publications

### Division de l'arborescence de rubriques entre plusieurs gestionnaires de files d'attente

Un gestionnaire de files d'attente d'hébergement de rubrique routé est uniquement responsable des connaissances d'abonnement et des messages de publication liés à la branche de l'arborescence de rubriques pour laquelle son objet de rubrique géré est configuré. Si différentes rubriques sont utilisées par différentes applications de publication / abonnement dans le cluster, vous pouvez configurer différents gestionnaires de files d'attente pour héberger différentes branches en cluster de l'arborescence de rubriques. Cela permet la mise à l'échelle en réduisant le trafic de publication, la connaissance des abonnements et les canaux sur chaque gestionnaire de files d'attente hôte de rubrique dans le cluster. Vous devez utiliser cette méthode pour les branches de volume élevé distinctes de l'arborescence de rubriques:

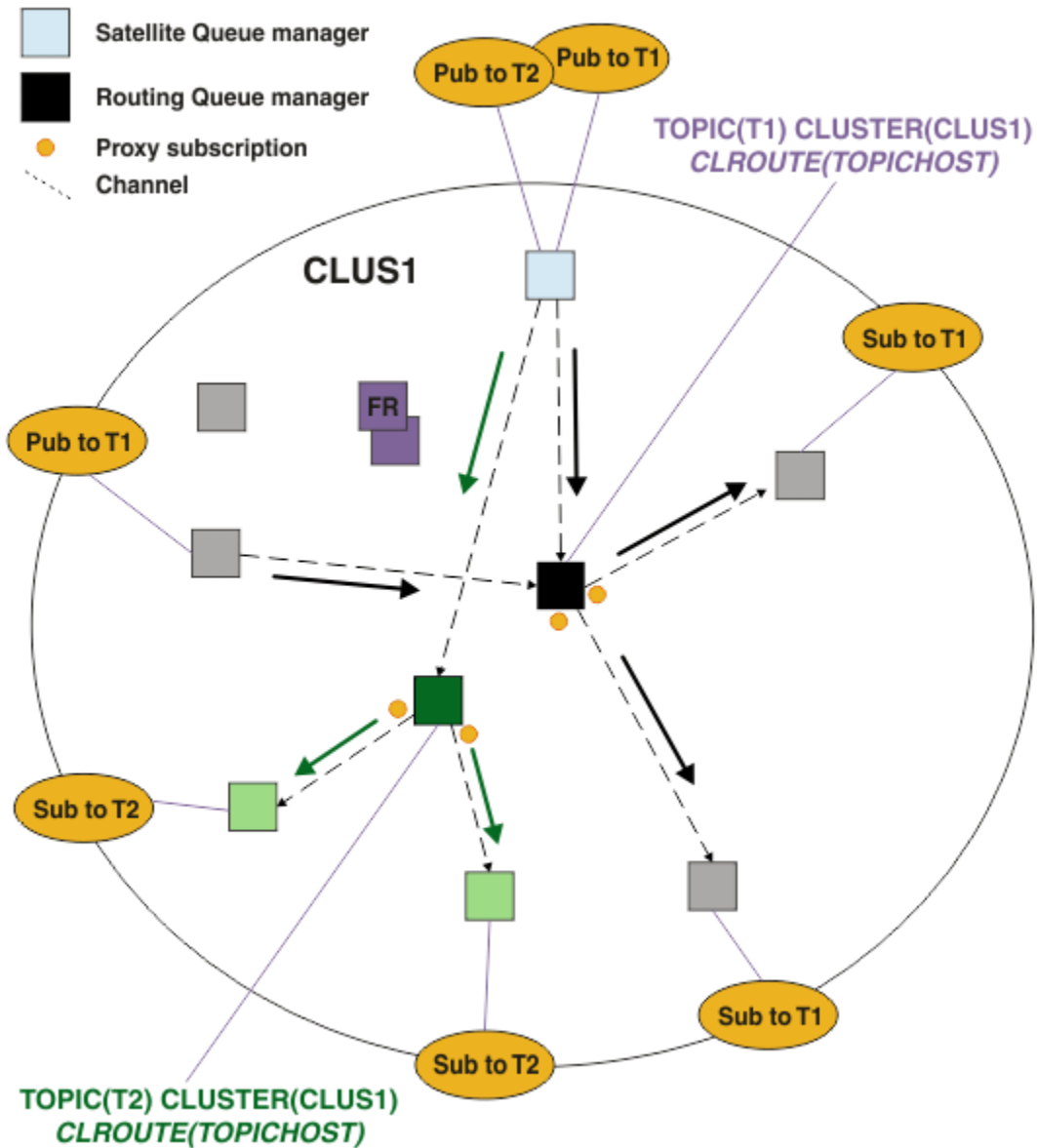


Figure 24. Un cluster de publication / abonnement routé par un hôte de rubrique avec deux rubriques, chacune définie sur un hôte de rubrique

Par exemple, en utilisant les rubriques décrites dans [Arborescences de rubriques](#), si la rubrique T1 a été configurée avec une chaîne de rubrique /USA/Alabama et la rubrique T2 a été configurée avec une chaîne de rubrique /USA/Alaska, un message publié dans /USA/Alabama/Mobile sera acheminé via le gestionnaire de files d'attente hébergeant T1, et un message publié dans /USA/Alaska/Juneau serait acheminé via le gestionnaire de files d'attente hébergeant T2.

**Remarque :** Vous ne pouvez pas créer un abonnement unique couvrant plusieurs branches en cluster de l'arborescence de rubriques en utilisant un caractère générique plus haut dans l'arborescence de rubriques que les points qui sont en cluster. Voir [Abonnements à des caractères génériques](#).

### Routage d'hôte de rubrique à l'aide de plusieurs hôtes de rubrique pour une même rubrique

Si un seul gestionnaire de files d'attente est responsable du routage d'une rubrique et que ce gestionnaire de files d'attente devient indisponible ou incapable de gérer la charge de travail, les publications ne seront pas acheminées rapidement vers les abonnements.

Si vous avez besoin d'une résilience, d'une évolutivité et d'un équilibrage de charge plus importants que ceux que vous obtenez lorsque vous définissez une rubrique sur un seul gestionnaire de files d'attente, vous pouvez définir une rubrique sur plusieurs gestionnaires de files d'attente. Chaque message publié est acheminé via un hôte de rubrique unique. Lorsqu'il existe plusieurs définitions d'hôte de rubrique correspondantes, l'un des hôtes de rubrique est choisi. Le choix est effectué de la même manière que pour les files d'attente en cluster. Cela permet aux messages d'être routés vers les hôtes de rubrique disponibles, en évitant ceux qui ne sont pas disponibles, et permet d'équilibrer la charge de travail entre plusieurs gestionnaires de files d'attente d'hôte de rubrique et canaux. Toutefois, l'ordre dans plusieurs messages n'est pas conservé lorsque vous utilisez plusieurs hôtes de rubrique pour la même rubrique dans le cluster.

Le diagramme suivant illustre un cluster routé par un hôte de rubrique dans lequel la même rubrique a été définie sur deux gestionnaires de files d'attente. Dans cet exemple, les gestionnaires de files d'attente d'abonnement envoient des informations sur la rubrique souscrite aux deux gestionnaires de files d'attente hôte de rubrique sous la forme d'un abonnement de proxy:

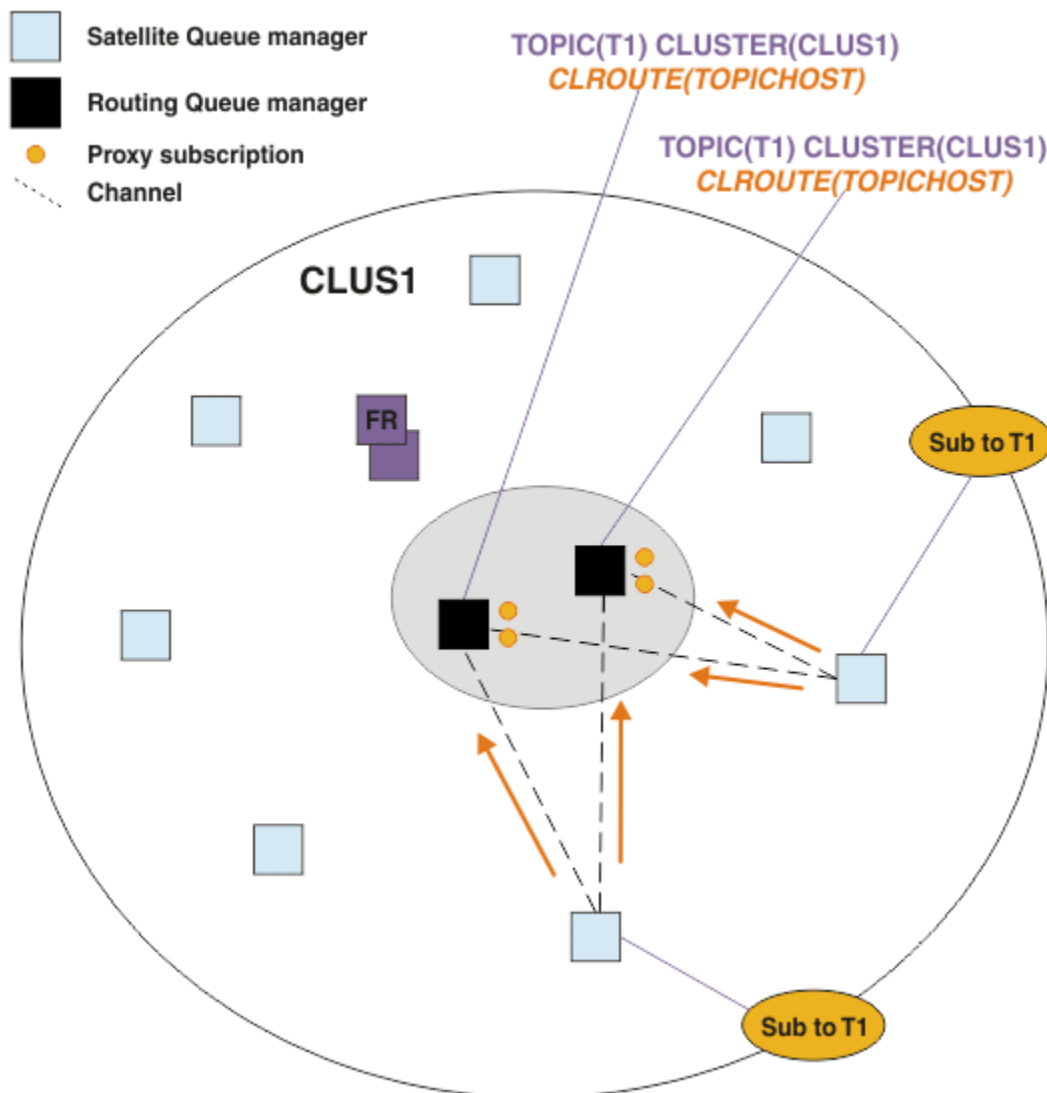


Figure 25. Création d'abonnements de proxy dans un cluster de publication / abonnement à plusieurs hôtes de rubriques

Lorsqu'une publication est effectuée à partir d'un gestionnaire de files d'attente non hôte, le gestionnaire de files d'attente envoie une copie de la publication à un des gestionnaires de files d'attente hôte de rubrique pour cette rubrique. Le système choisit l'hôte en fonction du comportement par défaut de l'algorithme de gestion de la charge de travail du cluster. Dans un système standard, cela correspond

approximativement à une distribution circulaire sur chaque gestionnaire de files d'attente hôte de rubrique. Il n'y a pas d'affinité entre les messages de la même application de publication ; cela revient à utiliser un type de liaison de cluster NOTFIXED.

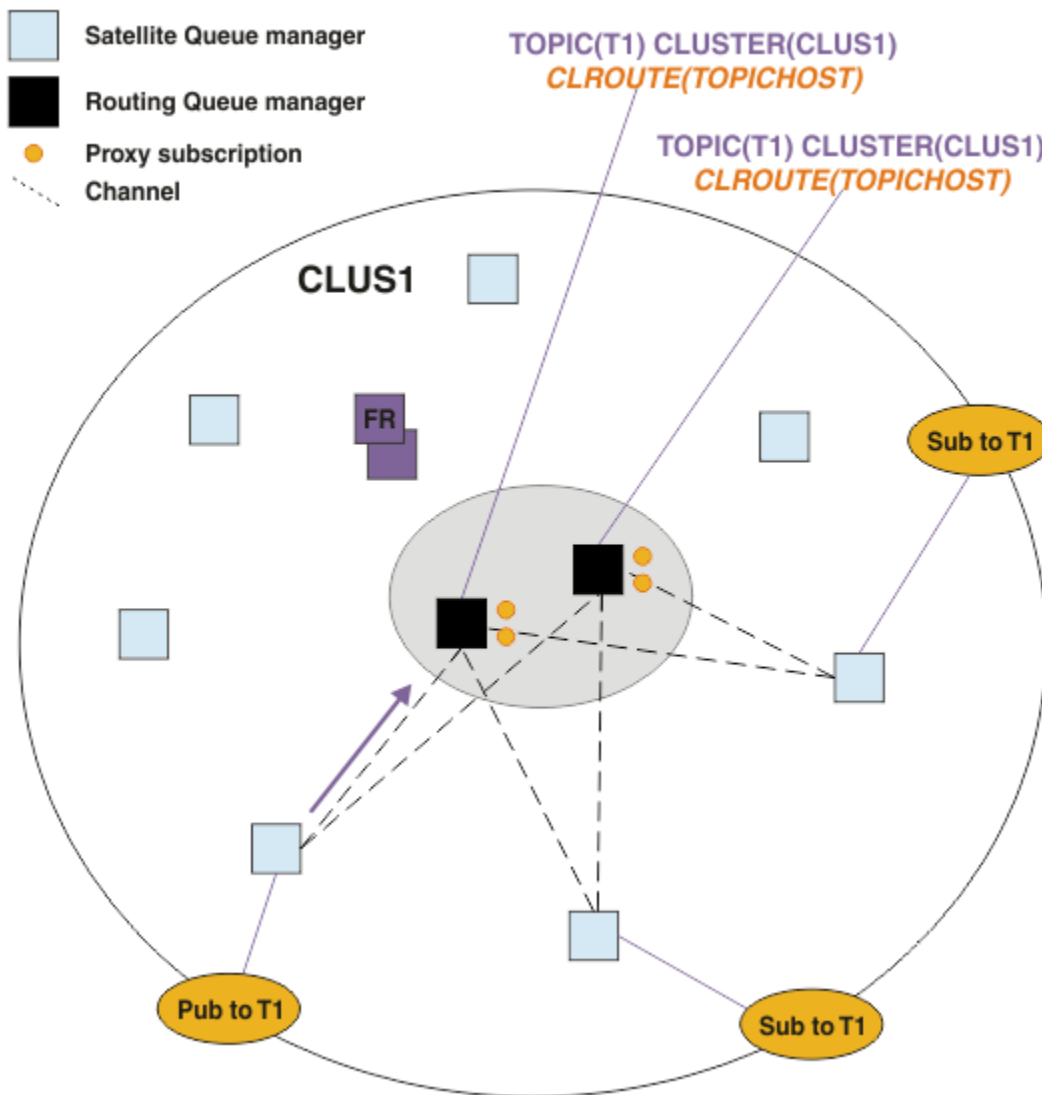


Figure 26. Réception de publications dans un cluster de publication / abonnement à plusieurs hôtes de rubriques

Les publications entrantes dans le gestionnaire de files d'attente hôte de rubrique choisi sont ensuite réacheminées vers tous les gestionnaires de files d'attente qui ont enregistré un abonnement proxy correspondant:



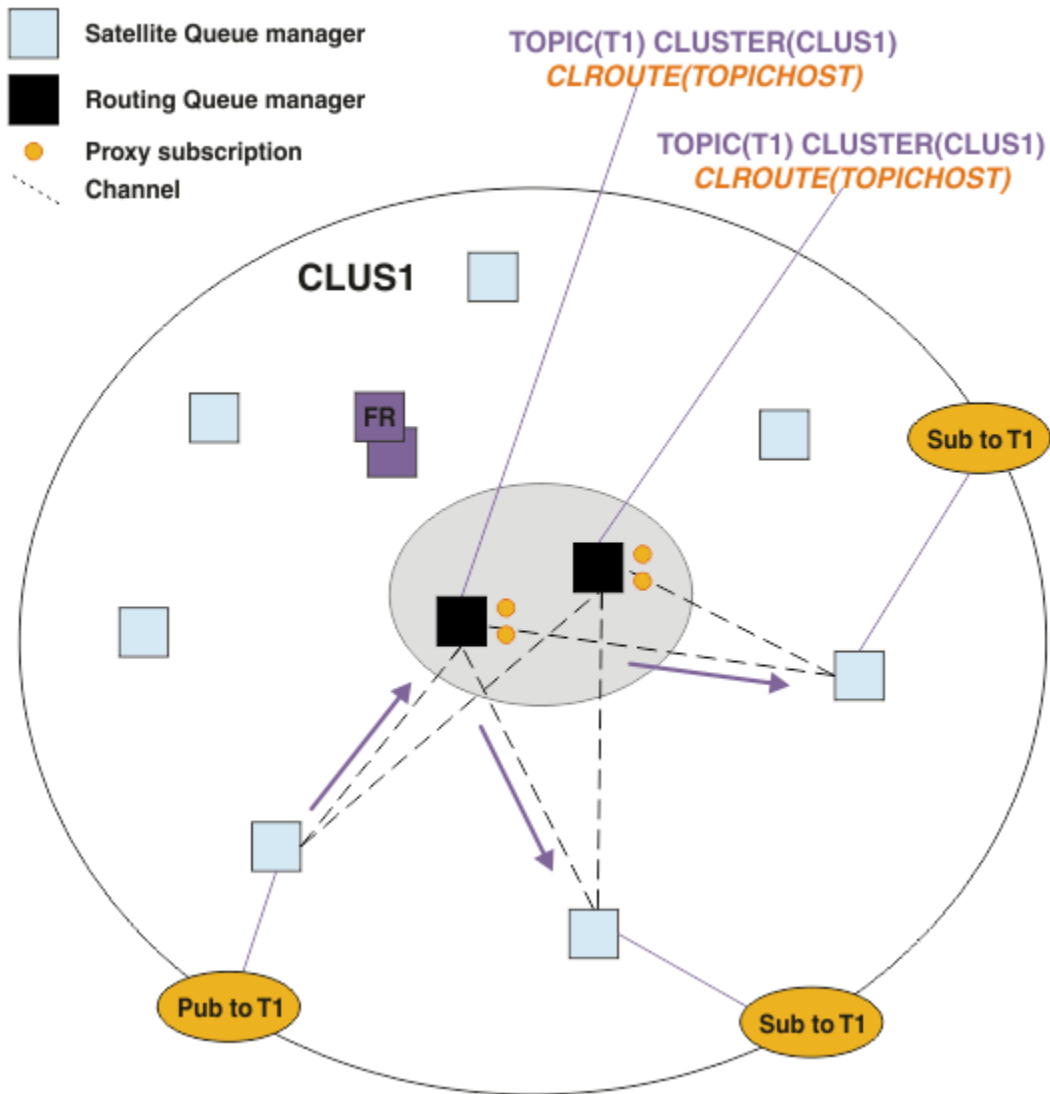


Figure 27. Routage des publications aux abonnés dans un cluster de publication / abonnement à plusieurs hôtes de rubriques

### Création d'abonnements et de diffuseurs de publications en local sur un gestionnaire de files d'attente hôte de rubrique

Les exemples ci-dessus montrent le routage entre les diffuseurs et les abonnés sur les gestionnaires de files d'attente qui n'hébergent pas d'objets de rubrique routés administrés. Dans ces topologies, les messages nécessitent plusieurs *tronçons* pour atteindre les abonnements.

Lorsque le tronçon supplémentaire n'est pas souhaitable, il peut être approprié de connecter des diffuseurs de publications de clés à des gestionnaires de files d'attente d'hébergement de rubriques. Toutefois, s'il existe plusieurs hôtes de rubrique pour une rubrique et un seul diffuseur de publications, tout le trafic de publication est acheminé via le gestionnaire de files d'attente hôte de rubrique auquel le diffuseur de publications est connecté.

De même, s'il existe des abonnements clés, ils peuvent se trouver sur un gestionnaire de files d'attente hôte de rubrique. Toutefois, s'il existe plusieurs hôtes de la rubrique routée, seule une partie des publications évitera le tronçon supplémentaire, le reste étant acheminé via les autres gestionnaires de files d'attente hôte de rubrique en premier.

Les topologies de ce type sont décrites plus en détail ici: [Routage de l'hôte de rubrique à l'aide de diffuseurs de publications ou d'abonnés centralisés.](#)

**Remarque :** Une planification spéciale est nécessaire si vous modifiez la configuration de rubrique routée lors de la colocalisation de diffuseurs ou d'abonnements avec des hôtes de rubrique routés. Par exemple, voir Ajout d'hôtes de rubrique supplémentaires à un cluster routé par un hôte de rubrique.

## Résumé et remarques supplémentaires

Un cluster de publication / abonnement routé par un hôte de rubrique vous donne un contrôle précis sur les gestionnaires de files d'attente qui hébergent chaque rubrique, et ces gestionnaires de files d'attente deviennent les gestionnaires de files d'attente de *routage* pour cette branche de l'arborescence de rubriques. En outre, les gestionnaires de files d'attente sans abonnements ou les diffuseurs de publications n'ont pas besoin de se connecter aux gestionnaires de files d'attente de l'hôte de rubrique, et les gestionnaires de files d'attente avec des abonnements n'ont pas besoin de se connecter aux gestionnaires de files d'attente qui n'hébergent pas de rubrique. Cette configuration peut réduire de manière significative le nombre de connexions entre les gestionnaires de files d'attente du cluster et la quantité d'informations transmises entre les gestionnaires de files d'attente. Cela est particulièrement vrai dans les grands clusters où seul un sous-ensemble de gestionnaires de files d'attente effectue un travail de publication / abonnement. Cette configuration vous permet également de contrôler la charge sur les gestionnaires de files d'attente individuels du cluster, de sorte que (par exemple) vous pouvez choisir d'héberger des rubriques très actives sur des systèmes plus puissants et plus résilients. Pour certaines configurations, notamment les clusters plus grands, il s'agit généralement d'une topologie plus appropriée que le *routage direct*.

Toutefois, le routage via un hôte de rubrique impose également certaines contraintes sur votre système :

- La configuration et la maintenance du système requièrent davantage de planification que dans le cas du routage direct. Vous devez choisir les points à regrouper dans l'arborescence de rubriques et l'emplacement des définitions de rubrique dans le cluster.
- Comme pour les rubriques routées directement, lorsqu'une nouvelle rubrique routée via un hôte de rubrique est définie, les informations sont envoyées aux gestionnaires de files d'attente des référentiels complets, et à partir de là, dirigées vers tous les membres du cluster. Cet événement déclenche le démarrage des canaux dans chaque membre du cluster depuis les référentiels complets s'ils ne sont pas encore démarrés.
- Les publications sont toujours envoyées à un gestionnaire de files d'attente hôte depuis un gestionnaire de files d'attente non hôte, même si le cluster ne comporte pas d'abonnement. Par conséquent, vous devez utiliser des rubriques routées lorsque des abonnements sont censés exister, ou lorsque le temps système de la connectivité globale et de la connaissance est supérieur au risque de trafic de publication supplémentaire.

**Remarque :** Comme décrit précédemment, le fait de rendre les diffuseurs de publications locaux à un hôte de sujet peut atténuer ce risque.

- Les messages qui sont publiés dans des gestionnaires de files d'attente non hôte ne sont pas envoyés directement au gestionnaire de files d'attente qui héberge l'abonnement ; ils sont toujours routés via un gestionnaire de files d'attente hôte de rubrique. Cette approche peut augmenter le temps système total dans le cluster, augmenter le temps d'attente des messages et réduire les performances.

**Remarque :** Comme décrit précédemment, le fait de rendre les abonnements ou les diffuseurs de publications locaux à un hôte de rubrique peut atténuer ce risque.

- L'utilisation d'un gestionnaire de files d'attente hôte de rubrique unique introduit un point de défaillance unique pour tous les messages qui sont publiés dans une rubrique. Vous pouvez supprimer ce point de défaillance unique en définissant plusieurs hôtes de rubrique. Cependant, l'existence de plusieurs hôtes a un impact sur l'ordre des messages publiés tels qu'ils sont reçus par les abonnements.
- Une charge supplémentaire de messages est engendrée par les gestionnaires de files d'attente hôte de rubrique car le trafic de publication provenant de plusieurs gestionnaires de files d'attente doit être traité par ces gestionnaires de files d'attente. Cette charge peut être réduite : utilisez plusieurs hôtes de rubrique pour une rubrique unique (dans ce cas, l'ordre des messages n'est pas conservé), ou utilisez des gestionnaires de files d'attente différents pour héberger des rubriques routées pour différentes branches de l'arborescence de rubriques.

Avant d'utiliser le routage via un hôte de rubrique, explorez les approches alternatives détaillées dans [«Routage direct dans les clusters de publication / abonnement»](#), à la page 84 et [«Routage dans les hiérarchies de publication / abonnement»](#), à la page 115.

### ***Mise en cluster de publication / abonnement: meilleures pratiques***

L'utilisation de rubriques en cluster facilite l'extension du domaine de publication / abonnement entre les gestionnaires de files d'attente, mais peut entraîner des problèmes si les mécanismes et les implications ne sont pas parfaitement compris. Il existe deux modèles pour le partage d'informations et le routage des publications. Implémentez le modèle qui répond le mieux à vos besoins métier individuels et qui est le plus performant sur le cluster que vous avez choisi.

Les informations sur les meilleures pratiques dans les sections suivantes ne fournissent pas une solution unique, mais partagent plutôt des approches communes pour résoudre des problèmes communs. Il suppose que vous avez une connaissance de base des clusters IBM MQ et de la messagerie de publication / abonnement, et que vous êtes familiarisé avec les informations des [réseaux de publication / abonnement distribués](#) et [«Conception des clusters de publication/abonnement»](#), à la page 82.

Lorsque vous utilisez un cluster pour la messagerie point-à-point, chaque gestionnaire de files d'attente du cluster fonctionne en fonction du besoin de connaissance. En d'autres termes, il détecte uniquement les autres ressources de cluster, telles que les autres gestionnaires de files d'attente du cluster et les files d'attente en cluster, lorsque les applications qui s'y connectent demandent à les utiliser. Lorsque vous ajoutez une messagerie de publication / abonnement à un cluster, un niveau accru de partage d'informations et de connectivité entre les gestionnaires de files d'attente de cluster est introduit. Pour pouvoir suivre les meilleures pratiques pour les clusters de publication / abonnement, vous devez bien comprendre les implications de ce changement de comportement.

Pour vous permettre de générer la meilleure architecture, en fonction de vos besoins précis, il existe deux modèles de partage d'informations et de routage de publication dans les clusters de publication / abonnement: *routage direct* et *routage via un hôte de rubrique*. Pour faire le bon choix, vous devez comprendre les deux modèles et les différentes exigences que chaque modèle satisfait. Ces exigences sont abordées dans les sections suivantes, en association avec [«Planification de votre réseau de publication / abonnement distribué»](#), à la page 78:

- [«Raisons de limiter le nombre de gestionnaires de files d'attente de cluster impliqués dans l'activité de publication / abonnement»](#), à la page 99
- [«Comment choisir les rubriques à mettre en cluster»](#), à la page 100
- [«Comment dimensionner votre système»](#), à la page 101
- [«Emplacement de l'éditeur et de l'abonnement»](#), à la page 102
- [«Trafic de publication»](#), à la page 102
- [«Modification d'abonnement et chaînes de rubrique dynamiques»](#), à la page 103

### **Raisons de limiter le nombre de gestionnaires de files d'attente de cluster impliqués dans l'activité de publication / abonnement**

Il existe des considérations relatives à la capacité et aux performances lorsque vous utilisez la messagerie de publication / abonnement dans un cluster. Par conséquent, il est recommandé de prendre soigneusement en compte la nécessité d'une activité de publication / abonnement dans les gestionnaires de files d'attente et de la limiter au nombre de gestionnaires de files d'attente qui en ont besoin. Une fois que l'ensemble minimal de gestionnaires de files d'attente qui doivent publier et s'abonner à des rubriques est identifié, ils peuvent devenir membres d'un cluster qui ne contient que ces derniers et aucun autre gestionnaire de files d'attente.

Cette approche est particulièrement utile si vous disposez d'un cluster établi qui fonctionne déjà bien pour la messagerie point-à-point. Lorsque vous transformez un grand cluster existant en un cluster de publication / abonnement, il est préférable de créer initialement un cluster distinct pour le travail de publication / abonnement où les applications peuvent être essayées, plutôt que d'utiliser le cluster en cours. Vous pouvez utiliser un sous-ensemble de gestionnaires de files d'attente existants qui se trouvent déjà dans un ou plusieurs clusters point-à-point et faire de ce sous-ensemble des membres du

nouveau cluster de publication / abonnement. Toutefois, les gestionnaires de files d'attente de référentiel complet de votre nouveau cluster ne doivent pas être membres d'un autre cluster ; cela isole la charge supplémentaire des référentiels complets de cluster existants.

Si vous ne pouvez pas créer de nouveau cluster et que vous devez transformer un grand cluster existant en un cluster de publication / abonnement, n'utilisez pas de modèle de routage direct. Le modèle routé par l'hôte de rubrique est généralement plus performant dans les clusters plus grands, car il limite généralement le partage d'informations de publication / abonnement et la connectivité à l'ensemble des gestionnaires de files d'attente qui effectuent activement des travaux de publication / abonnement, en se concentrant sur les gestionnaires de files d'attente hébergeant les rubriques. La seule exception à cela est si une actualisation manuelle des informations d'abonnement est appelée sur un gestionnaire de files d'attente hébergeant une définition de rubrique, auquel cas le gestionnaire de files d'attente hôte de rubrique se connecte à chaque gestionnaire de files d'attente du cluster. Voir [Resynchronisation des abonnements de proxy](#).

Si vous établissez qu'un cluster ne peut pas être utilisé pour la publication / l'abonnement en raison de sa taille ou de sa charge actuelle, il est recommandé d'éviter que ce cluster ne soit transformé de manière inattendue en un cluster de publication / abonnement. Utilisez la propriété de gestionnaire de files d'attente **PSCLUS** pour arrêter quiconque ajoute une rubrique en cluster sur un gestionnaire de files d'attente du cluster. Voir [«Blocage de la publication / l'abonnement en cluster»](#), à la page 110.

## Comment choisir les rubriques à mettre en cluster

Il est important de choisir avec soin les rubriques qui sont ajoutées au cluster: plus l'arborescence de rubriques est élevée, plus leur utilisation est répandue. Cela peut entraîner la propagation d'un plus grand nombre d'informations d'abonnement et de publications que nécessaire. S'il existe plusieurs branches distinctes de l'arborescence de rubriques, où certaines doivent être regroupées et d'autres non, créez des objets de rubrique gérés à la racine de chaque branche qui doit être mise en cluster et ajoutez-les au cluster. Par exemple, si les branches /A, /B et /C ont besoin de la mise en cluster, définissez des objets de rubrique en cluster distincts pour chaque branche.

**Remarque :** Le système vous empêche d'imbriquer des définitions de rubrique en cluster dans l'arborescence de rubriques. Vous n'êtes autorisé à regrouper des rubriques qu'à un seul point de l'arborescence de rubriques pour chaque sous-branche. Par exemple, vous ne pouvez pas définir d'objets de rubrique en cluster pour /A et pour /A/B. L'imbrication de rubriques en cluster peut entraîner une confusion quant à l'objet en cluster qui s'applique à tel ou tel abonnement, en particulier lorsque les abonnements utilisent des caractères génériques. Cela est encore plus important lorsque vous utilisez le routage d'hôte de sujet, où les décisions de routage sont précisément définies par votre allocation d'hôtes de sujet.

Si des rubriques en cluster doivent être ajoutées en haut de l'arborescence de rubriques, mais que certaines branches de l'arborescence sous le point en cluster ne nécessitent pas le comportement en cluster, vous pouvez utiliser les attributs de portée d'abonnement et de publication pour réduire le niveau d'abonnement et de partage de publication pour d'autres rubriques.

Vous ne devez pas placer le noeud racine de rubrique dans le cluster sans tenir compte du comportement observé. Rendez les rubriques globales évidentes lorsque cela est possible, par exemple en utilisant un qualificatif de haut niveau dans la chaîne de rubrique: /global ou /cluster.

Il existe une autre raison pour laquelle vous ne souhaitez pas que le noeud de rubrique racine soit mis en cluster. En effet, chaque gestionnaire de files d'attente possède une définition locale pour le noeud racine, l'objet de rubrique SYSTEM.BASE.TOPIC. Lorsque cet objet est mis en cluster sur un gestionnaire de files d'attente du cluster, tous les autres gestionnaires de files d'attente en sont informés. Toutefois, lorsqu'une définition locale du même objet existe, ses propriétés remplacent l'objet de cluster. Ainsi, les gestionnaires de files d'attente agissent comme si la rubrique n'était pas mise en cluster. Pour résoudre ce problème, vous devez mettre en cluster chaque définition de SYSTEM.BASE.TOPIC. Vous pouvez effectuer cette opération pour les définitions routées directement, mais pas pour les définitions routées d'hôte de rubrique, car chaque gestionnaire de files d'attente devient un hôte de rubrique.

## Comment dimensionner votre système

Les clusters de publication / abonnement génèrent généralement un modèle de canaux de cluster différent de la messagerie point-à-point dans un cluster. Le modèle point-à-point est un "opt-in", mais les clusters de publication / abonnement ont un caractère plus indiscriminé avec une distribution d'abonnement, en particulier lors de l'utilisation de rubriques acheminées directement. Par conséquent, il est important d'identifier les gestionnaires de files d'attente d'un cluster de publication / abonnement qui utiliseront des canaux de cluster pour se connecter à d'autres gestionnaires de files d'attente et dans quelles circonstances.

Le tableau suivant répertorie l'ensemble type de canaux émetteurs et récepteurs de cluster attendus pour chaque gestionnaire de files d'attente d'un cluster de publication / abonnement en cours d'exécution normale, en fonction du rôle de gestionnaire de files d'attente dans le cluster de publication / abonnement.

Rôle de gestionnaire de files d'attente	Récepteurs de cluster directs	Expéditeurs de cluster direct	Récepteurs de cluster de rubriques	Expéditeurs de cluster de rubriques
Référentiel complet	AllQmgrs	AllQmgrs	AllQmgrs	AllQMgers
Hôte de définition de rubrique	Non applicable	Non applicable	AllSubs+AllPubs (1)	AllSubs (1)
Abonnements créés	AllPubs (1)	AllQMgers	AllHosts	AllHosts
Diffuseurs de publications connectés	AllSubs (1)	AllSubs (1)	AllHosts	AllHosts
Aucun diffuseur de publications ni abonné	AllSubs (1)	Aucun (1)	Aucun (2)	Aucun (2)

### Clé :

#### AllQmgrs

Un canal vers et depuis chaque gestionnaire de files d'attente du cluster.

#### AllSubs

Un canal vers et depuis chaque gestionnaire de files d'attente où un abonnement a été créé.

#### AllPubs

Un canal vers et depuis chaque gestionnaire de files d'attente où une application de publication a été connectée.

#### AllHosts

Canal vers et depuis chaque gestionnaire de files d'attente dans lequel une définition de l'objet de rubrique en cluster a été configurée.

#### Aucun

Aucun canal vers ou depuis d'autres gestionnaires de files d'attente du cluster dans le seul but de la messagerie de publication / abonnement.

### Remarques :

1. Si une régénération de gestionnaire de files d'attente des abonnements de proxy est effectuée à partir de ce gestionnaire de files d'attente, un canal vers et depuis tous les autres gestionnaires de files d'attente du cluster peut être créé automatiquement.
2. Si une régénération de gestionnaire de files d'attente des abonnements de proxy est effectuée à partir de ce gestionnaire de files d'attente, un canal vers et à partir de tout autre gestionnaire de files d'attente du cluster qui héberge une définition d'une rubrique en cluster peut être automatiquement créé.

Le tableau précédent montre que le routage via un hôte de rubrique utilise généralement beaucoup moins de canaux émetteurs et récepteurs de cluster que le routage direct. Si la connectivité des canaux est une préoccupation pour certains gestionnaires de files d'attente dans un cluster, pour des raisons de capacité ou de capacité à établir certains canaux (par exemple, via des pare-feux), le routage via l'hôte de rubrique est donc une solution préférée.

## Emplacement de l'éditeur et de l'abonnement

La fonction de publication / abonnement en cluster permet aux messages publiés sur un gestionnaire de files d'attente d'être distribués aux abonnements sur n'importe quel autre gestionnaire de files d'attente du cluster. Comme pour la messagerie point-à-point, le coût de la transmission de messages entre les gestionnaires de files d'attente peut être préjudiciable aux performances. Par conséquent, vous devez envisager de créer des abonnements à des rubriques sur les mêmes gestionnaires de files d'attente que lorsque des messages sont publiés.

Lorsque vous utilisez le routage via un hôte de rubrique dans un cluster, il est important de prendre également en compte l'emplacement des abonnements et des diffuseurs de publications par rapport aux gestionnaires de files d'attente d'hébergement de rubrique. Lorsque le diffuseur de publications n'est pas connecté à un gestionnaire de files d'attente qui est un hôte de la rubrique en cluster, les messages publiés sont toujours envoyés à une rubrique hébergeant le gestionnaire de files d'attente. De même, lorsqu'un abonnement est créé sur un gestionnaire de files d'attente qui n'est pas un hôte de rubrique pour une rubrique en cluster, les messages publiés à partir d'autres gestionnaires de files d'attente du cluster sont toujours envoyés à un gestionnaire de files d'attente d'hébergement de rubrique en premier. Plus spécifiquement, si l'abonnement se trouve sur un gestionnaire de files d'attente qui héberge la rubrique, mais qu'il existe un ou plusieurs autres gestionnaires de files d'attente qui hébergent également la même rubrique, une proportion de publications provenant d'autres gestionnaires de files d'attente est acheminée via ces autres gestionnaires de files d'attente qui hébergent la rubrique. Pour plus d'informations sur la conception d'un cluster de publication / abonnement routé par un hôte de rubrique afin de réduire la distance entre les diffuseurs de publications et les abonnements, voir [Routage des hôtes de rubrique à l'aide de diffuseurs de publications ou d'abonnés centralisés](#).

## Trafic de publication

Les messages publiés par une application connectée à un gestionnaire de files d'attente dans un cluster sont transmis aux abonnements sur d'autres gestionnaires de files d'attente à l'aide de canaux émetteurs de cluster.

Lorsque vous utilisez le routage direct, les messages publiés prennent le chemin le plus court entre les gestionnaires de files d'attente. C'est-à-dire qu'ils vont directement du gestionnaire de files d'attente de publication vers chacun des gestionnaires de files d'attente avec des abonnements. Les messages ne sont pas transmis aux gestionnaires de files d'attente qui ne disposent pas d'abonnements pour la rubrique. Voir [Abonnements de proxy dans un réseau de publication / abonnement](#).

Lorsque le débit des messages de publication entre un gestionnaire de files d'attente et un autre dans le cluster est élevé, l'infrastructure du canal de cluster entre ces deux points doit être en mesure de maintenir le débit. Cela peut impliquer l'optimisation des canaux et de la file d'attente de transmission utilisés.

Lorsque vous utilisez le routage via un hôte de rubrique, chaque message publié sur un gestionnaire de files d'attente qui n'est pas un hôte de rubrique est transmis à un gestionnaire de files d'attente hôte de rubrique. Cela est indépendant du fait qu'un ou plusieurs abonnements existent ailleurs dans le cluster. Cela introduit d'autres facteurs à prendre en compte lors de la planification:

- Le temps d'attente supplémentaire de l'envoi préalable de chaque publication à un gestionnaire de files d'attente hôte de rubrique est-il acceptable?
- Chaque gestionnaire de files d'attente d'hôte de rubrique peut-il supporter le débit de publication entrant et sortant? Prenons l'exemple d'un système avec des diffuseurs de publications sur de nombreux gestionnaires de files d'attente différents. S'ils envoient tous leurs messages à un très petit ensemble de gestionnaires de files d'attente d'hébergement de rubrique, ces hôtes de rubrique

peuvent devenir un goulot d'étranglement dans le traitement de ces messages et leur routage vers les gestionnaires de files d'attente d'abonnement.

- Est-il prévu qu'une proportion importante des messages publiés n'aient pas d'abonné correspondant? Si tel est le cas et que le débit de publication de ces messages est élevé, il peut être préférable de faire du gestionnaire de files d'attente du diffuseur de publications un hôte de rubrique. Dans ce cas, tout message publié dans lequel aucun abonnement n'existe dans le cluster ne sera pas transmis à d'autres gestionnaires de files d'attente.

Ces problèmes peuvent également être résolus en introduisant plusieurs hôtes de rubrique, afin de répartir la charge de publication entre eux:

- Lorsqu'il existe plusieurs rubriques distinctes, chacune avec une proportion du trafic de publication, envisagez de les héberger sur des gestionnaires de files d'attente différents.
- Si les rubriques ne peuvent pas être séparées sur des hôtes de rubrique différents, envisagez de définir le même objet de rubrique sur plusieurs gestionnaires de files d'attente. Il en résulte un équilibrage de la charge de travail des publications sur chacune d'entre elles pour le routage. Toutefois, cela n'est approprié que lorsque l'ordre des messages de publication n'est pas requis.

## Modification d'abonnement et chaînes de rubrique dynamiques

Une autre considération est l'effet sur les performances du système pour la propagation des abonnements de proxy. Généralement, un gestionnaire de files d'attente envoie un message d'abonnement de proxy à certains autres gestionnaires de files d'attente du cluster lorsque le premier abonnement pour une chaîne de rubrique en cluster spécifique (et pas seulement un objet de rubrique configuré) est créé sur ce gestionnaire de files d'attente. De même, un message de suppression d'abonnement de proxy est envoyé lorsque le dernier abonnement d'une chaîne de rubrique en cluster spécifique est supprimé.

Pour le routage direct, chaque gestionnaire de files d'attente avec des abonnements envoie ces abonnements de proxy à tous les autres gestionnaires de files d'attente du cluster. Pour le routage via un hôte de rubrique, chaque gestionnaire de files d'attente avec des abonnements envoie uniquement les abonnements de proxy à chaque gestionnaire de files d'attente qui héberge une définition pour cette rubrique en cluster. Par conséquent, avec le routage direct, plus il y a de gestionnaires de files d'attente dans le cluster, plus le temps système de gestion des abonnements de proxy est élevé. En revanche, avec le routage via un hôte de rubrique, le nombre de gestionnaires de files d'attente dans le cluster n'est pas un facteur.

Dans les deux modèles de routage, si une solution de publication / abonnement consiste en un grand nombre de chaînes de rubrique uniques auxquelles vous êtes abonné, ou si les rubriques d'un gestionnaire de files d'attente du cluster sont fréquemment abonnées et désabonnées, une surcharge importante est observée sur ce gestionnaire de files d'attente, due à la génération constante de messages distribuant et supprimant les abonnements de proxy. Avec le routage direct, cela est aggravé par la nécessité d'envoyer ces messages à chaque gestionnaire de files d'attente du cluster.

Si le taux de modification des abonnements est trop élevé pour être adapté, même au sein d'un système routé par un hôte de rubrique, voir [Performances des abonnements dans les réseaux de publication / abonnement](#) pour plus d'informations sur les moyens de réduire le temps système d'abonnement du proxy.

### ***Définition de 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.

Lorsque vous utilisez le routage direct, l'emplacement de la définition de rubrique en cluster n'affecte pas directement le comportement du système, car tous les gestionnaires de files d'attente du cluster utilisent la définition de rubrique de la même manière. Par conséquent, vous devez définir la rubrique sur tout gestionnaire de files d'attente qui sera membre du cluster tant que la rubrique est nécessaire, et qui se trouve sur un système suffisamment fiable pour être régulièrement en contact avec les gestionnaires de files d'attente de référentiel complet.

Lorsque vous utilisez le routage via un hôte de rubrique, l'emplacement de la définition de rubrique en cluster est très important, car d'autres gestionnaires de files d'attente du cluster créent des canaux pour ce gestionnaire de files d'attente et lui envoient des informations d'abonnement et des publications. Pour choisir le meilleur gestionnaire de files d'attente pour héberger la définition de rubrique, vous devez comprendre le routage via l'hôte de rubrique. Voir [«Routage de l'hôte de rubrique dans les clusters de publication / abonnement»](#), à la page 89.

Si vous disposez d'une rubrique en cluster et d'un objet de rubrique local, la rubrique locale est prioritaire. Voir [«Plusieurs définitions de rubrique de cluster du même nom»](#), à la page 106.

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

## Héritage de rubriques en cluster

En règle générale, les applications de publication et d'abonnement dans une topologie de publication / abonnement en cluster s'attendent à fonctionner de la même manière, quel que soit le gestionnaire de files d'attente du cluster auquel elles sont connectées. C'est pourquoi les objets de rubrique gérés en cluster sont propagés à chaque gestionnaire de files d'attente du cluster.

Un objet de rubrique géré hérite de son comportement d'autres objets de rubrique gérés plus haut dans l'arborescence de rubriques. Cet héritage se produit lorsqu'une valeur explicite n'a pas été définie pour un paramètre de rubrique.

Dans le cas d'une publication / d'un abonnement en cluster, il est important de prendre en compte cet héritage car il introduit la possibilité que les diffuseurs de publications et les abonnés se comportent différemment en fonction du gestionnaire de files d'attente auquel ils se connectent. Si un objet de rubrique en cluster laisse des paramètres hérités d'objets de rubrique plus élevés, la rubrique peut se comporter différemment sur les différents gestionnaires de files d'attente du cluster. De même, les objets de rubrique définis en local définis sous un objet de rubrique en cluster dans l'arborescence de rubriques signifient que ces rubriques inférieures sont toujours en cluster, mais l'objet local peut modifier son comportement d'une manière qui diffère des autres gestionnaires de files d'attente du cluster.

## Abonnements génériques

Les abonnements de proxy sont créés lorsque des abonnements locaux sont effectués à une chaîne de rubrique qui se résout au niveau ou en dessous d'un objet de rubrique en cluster. Si un abonnement générique est effectué plus haut dans la hiérarchie de rubriques que n'importe quelle rubrique de cluster, il n'est pas associé à des abonnements de proxy envoyés autour du cluster pour la rubrique de cluster correspondante et ne reçoit donc aucune publication des autres membres du cluster. Il reçoit toutefois les publications du gestionnaire de files d'attente local.

Toutefois, si une autre application s'abonne à une chaîne de rubrique qui se résout dans ou en dessous de la rubrique de cluster, des abonnements de proxy sont générés et des publications sont propagées à ce gestionnaire de files d'attente. A l'arrivée de l'original, l'abonnement générique plus élevé est considéré comme un destinataire légitime de ces publications et reçoit une copie. Si ce comportement n'est pas requis, définissez **WILDCARD (BLOCK)** sur la rubrique en cluster. Ainsi, le caractère générique d'origine



n'est pas considéré comme un abonnement légitime et il ne reçoit plus aucune publication (locale ou provenant d'un autre emplacement du cluster) sur la rubrique du cluster ou ses sous-rubriques.

### **Concepts associés**

[Utilisation des rubriques d'administration](#)

[Utilisation des abonnements](#)

### **Référence associée**

[DISPLAYTOPIC](#)

[STATUT D'AFFICHAGE](#)

[DISPLAYSUB](#)

#### *Attributs de rubrique de cluster*

Lorsqu'un objet de rubrique a l'attribut de nom de cluster défini, la définition de rubrique est propagée à tous les gestionnaires de files d'attente du cluster. Chaque gestionnaire de files d'attente utilise les attributs de rubrique propagés pour contrôler le comportement des applications de publication / abonnement.

Un objet de rubrique comporte un certain nombre d'attributs qui s'appliquent aux clusters de publication / abonnement. Certains contrôlent le comportement général des applications de publication et d'abonnement et d'autres contrôlent la manière dont la rubrique est utilisée dans le cluster.

Une définition d'objet de rubrique en cluster doit être configurée de sorte que tous les gestionnaires de files d'attente du cluster puissent l'utiliser correctement.

Par exemple, si les files d'attente modèle doivent être utilisées pour les abonnements gérés (MDURMDL et MNDURMDL) sont définis sur un nom de file d'attente autre que le nom par défaut, que la file d'attente modèle nommée doit être définie sur tous les gestionnaires de files d'attente dans lesquels des abonnements gérés seront créés.

De même, si un attribut est défini sur ASPARENT, le comportement de la rubrique dépend des noeuds supérieurs de l'arborescence de rubriques (voir [Objets de rubrique d'administration](#)). sur chaque gestionnaire de files d'attente individuel du cluster. Cela peut entraîner un comportement différent lors de la publication ou de l'abonnement à partir de différents gestionnaires de files d'attente.

Les principaux attributs qui sont directement liés au comportement de publication / abonnement dans le cluster sont les suivants:

### **CLROUTE**

Ce paramètre contrôle le routage des messages entre les gestionnaires de files d'attente où les diffuseurs sont connectés et les gestionnaires de files d'attente où des abonnements correspondants existent.

- Vous configurez la route pour qu'elle soit directe entre ces gestionnaires de files d'attente ou via un gestionnaire de files d'attente qui héberge une définition de la rubrique en cluster. Voir [Clusters de publication / abonnement](#) pour plus de détails.
- Vous ne pouvez pas modifier le paramètre **CLROUTE** tant que le paramètre **CLUSTER** est défini. Pour modifier le **CLROUTE**, définissez d'abord la propriété **CLUSTER** pour qu'elle soit vide. Cela empêche les applications qui utilisent la rubrique de se comporter de manière groupée. Cela entraîne à son tour une interruption des publications distribuées aux abonnements. Vous devez donc également mettre au repos la messagerie de publication / abonnement lors de la modification.

### **PROXYSUB**

Ce paramètre contrôle le moment où les abonnements de proxy sont effectués.

- **FIRSTUSE** est la valeur par défaut et entraîne l'envoi d'abonnements de proxy en réponse à des abonnements locaux sur un gestionnaire de files d'attente dans une topologie de publication / abonnement distribuée et leur annulation lorsqu'ils ne sont plus nécessaires. Pour plus de détails sur la raison pour laquelle vous pouvez modifier cet attribut à partir de la valeur par défaut **FIRSTUSE**, voir [Transfert d'abonnement de proxy individuel et \*publication partout\*](#).

- Pour activer la *publication partout*, définissez le paramètre **PROXYSUB** sur FORCE pour un objet de rubrique de haut niveau. Il en résulte un seul abonnement de proxy générique qui correspond à toutes les rubriques sous cet objet de rubrique dans l'arborescence de rubriques.

**Remarque :** La définition de l'attribut **PROXYSUB (FORCE)** dans un cluster de publication / abonnement volumineux ou occupé peut entraîner une charge excessive sur les ressources système. L'attribut **PROXYSUB (FORCE)** est propagé à chaque gestionnaire de files d'attente, et pas seulement au gestionnaire de files d'attente sur lequel la rubrique a été définie. Ainsi, chaque gestionnaire de files d'attente du cluster crée un abonnement de proxy générique.

Une copie d'un message dans cette rubrique, publiée sur n'importe quel gestionnaire de files d'attente du cluster, est envoyée à chaque gestionnaire de files d'attente du cluster, soit directement, soit via un gestionnaire de files d'attente hôte de rubrique, en fonction du paramètre **CLROUTE**.

Lorsque la rubrique est routée directement, chaque gestionnaire de files d'attente crée des canaux émetteurs de cluster vers tous les autres gestionnaires de files d'attente. Lorsque la rubrique est routée par l'hôte de rubrique, les canaux vers chaque gestionnaire de files d'attente hôte de rubrique sont créés à partir de chaque gestionnaire de files d'attente du cluster.

Pour plus d'informations sur le paramètre **PROXYSUB** lorsqu'il est utilisé dans des clusters, voir [Performances de publication / abonnement avec routage direct](#).

## **PUBSCOBE et SUBSCOPE**

Ces paramètres déterminent si ce gestionnaire de files d'attente propage des publications aux gestionnaires de files d'attente de la topologie (cluster ou hiérarchie de publication / abonnement) ou limite la portée à son gestionnaire de files d'attente local. Vous pouvez effectuer le travail équivalent à l'aide d'un programme à l'aide de MQPMO\_SCOPE\_QMGR et de MQSO\_SCOPE\_QMGR.

### **PUBSCOPE**

Si un objet de rubrique de cluster est défini avec **PUBSCOPE (QMGR)**, la définition est partagée avec le cluster, mais la portée des publications basées sur cette rubrique est locale uniquement et elles ne sont pas envoyées aux autres gestionnaires de files d'attente du cluster.

### **SUBSCOPE**

Si un objet de rubrique de cluster est défini avec **SUBSCOPE (QMGR)**, la définition est partagée avec le cluster, mais la portée des abonnements basés sur cette rubrique est locale uniquement. Par conséquent, aucun abonnement de proxy n'est envoyé aux autres gestionnaires de files d'attente du cluster.

Ces deux attributs sont généralement utilisés ensemble pour isoler un gestionnaire de files d'attente de l'interaction avec d'autres membres du cluster sur des sujets particuliers. Le gestionnaire de files d'attente ne publie ni ne reçoit de publications sur ces rubriques vers et depuis d'autres membres du cluster. Cette situation n'empêche pas la publication ou l'abonnement si des objets de rubrique sont définis sur des sous-rubriques.

La définition de **SUBSCOPE** sur QMGR dans une définition locale d'une rubrique n'empêche pas les autres gestionnaires de files d'attente du cluster de propager leurs abonnements de proxy au gestionnaire de files d'attente s'ils utilisent une version en cluster de la rubrique, avec **SUBSCOPE (ALL)**. Toutefois, si la définition locale définit également **PUBSCOPE** sur QMGR, ces abonnements de proxy ne sont pas envoyés à partir de ce gestionnaire de files d'attente.

## **Concepts associés**

[Portée de la publication](#)

[Portée de l'abonnement](#)

### *Plusieurs définitions de rubrique de cluster du même nom*

Vous pouvez définir le même objet de rubrique de cluster nommé sur plusieurs gestionnaires de files d'attente dans le cluster et, dans certains cas, cela permet un comportement spécifique. Lorsqu'il existe plusieurs définitions de rubrique de cluster portant le même nom, la majorité des propriétés doivent correspondre. Si ce n'est pas le cas, des erreurs ou des avertissements sont signalés en fonction de la signification de la non-concordance.

En général, en cas de non-concordance dans les propriétés de plusieurs définitions de rubrique de cluster, des avertissements sont émis et l'une des définitions d'objet de rubrique est utilisée par chaque gestionnaire de files d'attente du cluster. La définition utilisée par chaque gestionnaire de files d'attente n'est pas déterministe ni cohérente entre les gestionnaires de files d'attente du cluster. Ces inadéquations devraient être résolues le plus rapidement possible.

Lors de la configuration ou de la maintenance du cluster, vous devez parfois créer plusieurs définitions de rubrique de cluster qui ne sont pas identiques. Cependant, cette mesure n'est utile que comme mesure temporaire, et elle est donc considérée comme une condition d'erreur potentielle.

Lorsque des non-concordances sont détectées, les messages d'avertissement suivants sont consignés dans le journal des erreurs de chaque gestionnaire de files d'attente:

- **Multi** Sur [Multiplateformes](#), [AMQ9465](#) et [AMQ9466](#).
- **z/OS** Sous z/OS, [CSQX465I](#) et [CSQX466I](#).

Les propriétés choisies pour n'importe quelle chaîne de rubrique sur chaque gestionnaire de files d'attente peuvent être déterminées en affichant le statut de la rubrique plutôt que les définitions d'objet de rubrique, par exemple à l'aide de **DISPLAY TPSTATUS**.

Dans certaines situations, un conflit dans les propriétés de configuration est suffisamment grave pour arrêter la création de l'objet de rubrique ou pour que les objets non concordants soient marqués comme non valides et ne soient pas propagés dans le cluster (voir **CLSTATE** dans **DISPLAY TOPIC**). Ces situations se produisent en cas de conflit dans la propriété de routage de cluster (**CLROUTE**) des définitions de rubrique. En outre, en raison de l'importance de la cohérence entre les définitions routées par les hôtes de rubrique, d'autres incohérences sont rejetées comme indiqué dans les sections suivantes de cet article.

Si le conflit est détecté lors de la définition de l'objet, le changement de configuration est rejeté. S'ils sont détectés ultérieurement par les gestionnaires de files d'attente de référentiel complet, les messages d'avertissement suivants sont consignés dans les journaux d'erreurs des gestionnaires de files d'attente:

- **Multi** Sous [Multiplateformes](#): [AMQ9879](#)
- **z/OS** Sous z/OS: [CSQX879E](#).

Lorsque plusieurs définitions du même objet de rubrique sont définies dans le cluster, une définition définie localement est prioritaire sur toute définition définie à distance. Par conséquent, s'il existe des différences dans les définitions, les gestionnaires de files d'attente hébergeant les définitions multiples se comportent différemment les uns des autres.

## Effet de la définition d'une rubrique non-cluster portant le même nom qu'une rubrique de cluster à partir d'un autre gestionnaire de files d'attente

Il est possible de définir un objet de rubrique géré qui n'est pas mis en cluster sur un gestionnaire de files d'attente qui se trouve dans un cluster et de définir simultanément le même objet de rubrique nommé qu'une définition de rubrique en cluster sur un autre gestionnaire de files d'attente. Dans ce cas, l'objet de rubrique défini localement est prioritaire sur toutes les définitions distantes du même nom.

Cela a pour effet d'empêcher le comportement de mise en cluster de la rubrique lorsqu'elle est utilisée à partir de ce gestionnaire de files d'attente. En d'autres termes, il se peut que les abonnements ne reçoivent pas de publications des diffuseurs de publications distants et que les messages des diffuseurs de publications ne soient pas propagés aux abonnements distants du cluster.

Une attention particulière doit être accordée avant de configurer un tel système, car cela peut entraîner un comportement confus.

**Remarque :** Si un gestionnaire de files d'attente individuel doit empêcher la propagation des publications et des abonnements dans le cluster, même lorsque la rubrique a été mise en cluster ailleurs, une autre approche consiste à définir les portées de publication et d'abonnement uniquement sur le gestionnaire de files d'attente local. Voir «Attributs de rubrique de cluster», à la page 105.

## Définitions de rubrique de cluster multiples dans un cluster routé directement

Pour le routage direct, vous ne définissez généralement pas la même rubrique de cluster sur plusieurs gestionnaires de files d'attente de cluster. En effet, le routage direct rend la rubrique disponible sur tous les gestionnaires de files d'attente du cluster, quel que soit le gestionnaire de files d'attente sur lequel elle a été définie. En outre, l'ajout de plusieurs définitions de rubrique de cluster augmente considérablement l'activité du système et la complexité administrative, et avec une complexité accrue, il y a plus de risques d'erreur humaine:

- Chaque définition entraîne l'envoi d'un objet de rubrique de cluster supplémentaire aux autres gestionnaires de files d'attente du cluster, y compris les autres gestionnaires de files d'attente hôte de rubrique de cluster.
- Toutes les définitions d'une rubrique spécifique d'un cluster doivent être identiques, faute de quoi il est difficile de déterminer quelle définition de rubrique est utilisée par un gestionnaire de files d'attente.

Il n'est pas non plus essentiel que le seul gestionnaire de files d'attente hôte soit disponible en permanence pour que la rubrique fonctionne correctement dans le cluster, car la définition de rubrique de cluster est mise en cache par les gestionnaires de files d'attente de référentiel complet et par tous les autres gestionnaires de files d'attente dans leurs référentiels de cluster partiels. Pour plus d'informations, voir [Disponibilité des gestionnaires de files d'attente hôte de rubrique qui utilisent le routage direct](#).

Pour une situation dans laquelle vous devrez peut-être définir temporairement une rubrique de cluster sur un deuxième gestionnaire de files d'attente, par exemple lorsque l'hôte existant de la rubrique doit être supprimé du cluster, voir [Déplacement d'une définition de rubrique de cluster vers un autre gestionnaire de files d'attente](#).

Si vous devez modifier une définition de rubrique de cluster, veillez à la modifier dans le même gestionnaire de files d'attente que dans celui dans lequel il a été défini. Une tentative de modification à partir d'un autre gestionnaire de files d'attente peut créer accidentellement une deuxième définition de la rubrique avec des attributs de rubrique en conflit.

## Définitions de rubrique de cluster multiples dans un cluster routé via un hôte de rubrique

Lorsqu'une rubrique de cluster est définie avec une route de cluster de l' *hôte de rubrique*, la rubrique est propagée sur tous les gestionnaires de files d'attente du cluster de la même manière que pour les rubriques acheminées *directement*. De plus, tous les messages de publication / abonnement de cette rubrique sont acheminés via les gestionnaires de files d'attente dans lesquels cette rubrique est définie. Par conséquent, l'emplacement et le nombre de définitions de la rubrique dans le cluster deviennent importants (voir «[Routage de l'hôte de rubrique dans les clusters de publication / abonnement](#)», à la page 89).

Pour garantir une disponibilité et une évolutivité adéquates, il est utile, si possible, de disposer de plusieurs définitions de rubrique. Voir [Disponibilité des gestionnaires de files d'attente hôte de rubrique qui utilisent le routage via un hôte de rubrique](#).

Lors de l'ajout ou de la suppression de définitions supplémentaires d'une rubrique routée *hôte de rubrique* dans un cluster, vous devez prendre en compte le flux de messages au moment du changement de configuration. Si des messages sont publiés dans le cluster dans la rubrique au moment de la modification, un processus de transfert est requis pour ajouter ou supprimer une définition de rubrique. Voir [Déplacement d'une définition de rubrique de cluster vers un autre gestionnaire de files d'attente et Ajout d'hôtes de rubrique supplémentaires à un cluster routé par un hôte de rubrique](#).

Comme expliqué précédemment, les propriétés des définitions multiples doivent correspondre, à l'exception éventuelle du paramètre **PUB**, comme décrit dans la section suivante. Lorsque les publications sont acheminées via des gestionnaires de files d'attente d'hôte de rubrique, il est encore plus important que plusieurs définitions soient cohérentes. Par conséquent, une incohérence détectée dans la chaîne de rubrique ou le nom de cluster est rejetée si une ou plusieurs des définitions de rubrique ont été configurées pour le routage du cluster d'hôtes de rubrique.

**Remarque :** Les définitions de rubrique de cluster sont également rejetées si une tentative est effectuée pour les configurer au-dessus ou en dessous d'une autre rubrique de l'arborescence de rubriques, où la définition de rubrique de cluster existante est configurée pour le routage via un hôte de rubrique. Cela permet d'éviter toute ambiguïté dans le routage des publications par rapport aux abonnements utilisant des caractères génériques.

## Traitement spécial pour le paramètre PUB

Le paramètre **PUB** permet de contrôler le moment où les applications peuvent publier dans une rubrique. Dans le cas du routage via un hôte de rubrique dans un cluster, il peut également contrôler les gestionnaires de files d'attente d'hôte de rubrique utilisés pour acheminer les publications. Pour cette raison, il est autorisé d'avoir plusieurs définitions du même objet de rubrique dans le cluster, avec des valeurs différentes pour le paramètre PUB .

Si plusieurs définitions de cluster distant d'une rubrique ont des valeurs différentes pour ce paramètre, la rubrique autorise l'envoi et la distribution des publications aux abonnements si les conditions suivantes sont remplies:

- Il n'existe pas d'objet de rubrique correspondant défini sur le gestionnaire de files d'attente auquel le diffuseur de publications est connecté et qui est défini sur PUB (DISABLED) .
- Une ou plusieurs des définitions de rubrique du cluster sont définies sur PUB (ENABLED) , ou une ou plusieurs des définitions de rubrique sont définies sur PUB (ASPARENT) et les gestionnaires de files d'attente locaux sur lesquels le diffuseur de publications est connecté et l'abonnement défini sont définis sur PUB (ENABLED) à un point supérieur de l'arborescence de rubriques.

Pour le routage via un hôte de rubrique, lorsque des messages sont publiés par des applications connectées à des gestionnaires de files d'attente qui ne sont pas des hôtes de rubrique, les messages sont uniquement acheminés vers les gestionnaires de files d'attente d'hébergement de rubrique où le paramètre **PUB** n'a pas été explicitement défini sur DISABLED. Vous pouvez donc utiliser le paramètre PUB (DISABLED) pour mettre au repos le trafic des messages via certains hôtes de rubrique. Vous pouvez effectuer cette opération pour préparer la maintenance ou la suppression d'un gestionnaire de files d'attente ou pour les raisons décrites dans la rubrique [Ajout d'hôtes de rubrique supplémentaires à un cluster routé par un hôte de rubrique](#).

### *Disponibilité des gestionnaires de files d'attente hôte de rubrique en cluster*

Concevez votre cluster de publication / abonnement pour réduire le risque que, si un gestionnaire de files d'attente hôte de rubrique devient indisponible, le cluster ne puisse plus traiter le trafic de la rubrique. L'effet de l'indisponibilité d'un gestionnaire de files d'attente d'hôte de rubrique varie selon que le cluster utilise le routage d'hôte de rubrique ou le routage direct.

## Disponibilité des gestionnaires de files d'attente hôte de rubrique qui utilisent le routage direct

Pour le routage direct, vous ne définissez généralement pas la même rubrique de cluster sur plusieurs gestionnaires de files d'attente de cluster. En effet, le routage direct rend la rubrique disponible sur tous les gestionnaires de files d'attente du cluster, quel que soit le gestionnaire de files d'attente sur lequel elle a été définie. Voir [Plusieurs définitions de rubrique de cluster dans un cluster routé directement](#).

Dans un cluster, chaque fois que l'hôte d'un objet en cluster (par exemple, une file d'attente en cluster ou une rubrique en cluster) devient indisponible pendant une période prolongée, les autres membres du cluster finissent par expirer la connaissance de ces objets. Dans le cas d'une rubrique en cluster, si le gestionnaire de files d'attente hôte de rubrique de cluster devient indisponible, les autres gestionnaires de files d'attente continuent de traiter les demandes de publication / abonnement de la rubrique de manière directe en cluster (c'est-à-dire, en envoyant des publications à des abonnements sur des gestionnaires de files d'attente éloignés) pendant au moins 60 jours à compter de la date à laquelle le gestionnaire de files d'attente d'hébergement de rubrique a été mis en communication pour la dernière fois avec les gestionnaires de files d'attente de référentiel complet. Si le gestionnaire de files d'attente sur lequel vous avez défini l'objet de rubrique de cluster n'est plus jamais disponible, les objets de rubrique mis en cache sur les autres gestionnaires de files d'attente sont supprimés et la rubrique revient à

une rubrique locale, auquel cas les abonnements cessent de recevoir des publications des applications connectées aux gestionnaires de files d'attente éloignées.

Avec la période de 60 jours pour récupérer le gestionnaire de files d'attente sur lequel vous définissez un objet de rubrique de cluster, il est peu nécessaire de prendre des mesures spéciales pour garantir qu'un hôte de rubrique de cluster reste disponible (notez, toutefois, que tous les abonnements définis sur l'hôte de rubrique de cluster non disponible ne restent pas disponibles). Le délai de 60 jours est suffisant pour faire face à des problèmes techniques et risque d'être dépassé uniquement en raison d'erreurs administratives. Pour limiter cette possibilité, si l'hôte de rubrique de cluster n'est pas disponible, tous les membres du cluster écrivent des messages de journal des erreurs toutes les heures, indiquant que leur objet de rubrique de cluster en cache n'a pas été actualisé. Répondez à ces messages en vous assurant que le gestionnaire de files d'attente sur lequel l'objet de rubrique de cluster est défini est en cours d'exécution. S'il n'est pas possible de rendre le gestionnaire de files d'attente hôte de rubrique de cluster à nouveau disponible, définissez la même définition de rubrique de cluster, avec exactement les mêmes attributs, sur un autre gestionnaire de files d'attente du cluster.

## **Disponibilité des gestionnaires de files d'attente d'hôte de rubrique qui utilisent le routage d'hôte de rubrique**

Pour le routage via un hôte de rubrique, toutes les messages de publication / abonnement d'une rubrique sont acheminés via les gestionnaires de files d'attente dans lesquels cette rubrique est définie. Pour cette raison, il est très important de prendre en compte la disponibilité continue de ces gestionnaires de files d'attente dans le cluster. Si un hôte de rubrique devient indisponible et qu'aucun autre hôte n'existe pour la rubrique, le trafic des diffuseurs vers les abonnés des différents gestionnaires de files d'attente du cluster s'arrête immédiatement pour la rubrique. Si des hôtes de rubrique supplémentaires sont disponibles, les gestionnaires de files d'attente de cluster acheminent le nouveau trafic de publication via ces hôtes de rubrique, fournissant ainsi une disponibilité continue des routes de message.

Comme pour les rubriques directes, après 60 jours, si le premier hôte de rubrique est toujours indisponible, la connaissance de la rubrique de cet hôte de rubrique est supprimée du cluster. S'il s'agit de la dernière définition restante pour cette rubrique dans le cluster, tous les autres gestionnaires de files d'attente cessent de transmettre des publications à n'importe quel hôte de rubrique pour le routage.

Pour garantir une disponibilité et une évolutivité adéquates, il est donc utile, si possible, de définir chaque rubrique sur au moins deux gestionnaires de files d'attente de cluster. Cela offre une protection contre l'indisponibilité d'un gestionnaire de files d'attente hôte de rubrique donné. Voir aussi [Plusieurs définitions de rubrique de cluster dans un cluster routé par un hôte de rubrique](#).

Si vous ne pouvez pas configurer plusieurs hôtes de rubrique (par exemple, parce que vous devez conserver l'ordre des messages) et que vous ne pouvez pas configurer un seul hôte de rubrique (car la disponibilité d'un seul gestionnaire de files d'attente ne doit pas affecter le flux des publications vers les abonnements dans tous les gestionnaires de files d'attente du cluster), envisagez de configurer la rubrique en tant que rubrique routée directement. Cela évite de dépendre d'un seul gestionnaire de files d'attente pour l'ensemble du cluster, mais nécessite que chaque gestionnaire de files d'attente individuel soit disponible pour qu'il puisse traiter les abonnements et les diffuseurs hébergés localement.

### ***Blocage de la publication / l'abonnement en cluster***

L'introduction de la première rubrique en cluster routée directement dans un cluster force chaque gestionnaire de files d'attente du cluster à prendre connaissance de tous les autres gestionnaires de files d'attente et peut les amener à créer des canaux entre eux. Si cela n'est pas souhaitable, vous devez à la place configurer la publication / l'abonnement routé par l'hôte de rubrique. Si l'existence d'une rubrique de cluster routée directement risque de compromettre la stabilité du cluster, en raison de problèmes de mise à l'échelle de chaque gestionnaire de files d'attente, vous pouvez désactiver complètement la fonctionnalité de publication / abonnement en cluster en définissant **PSCLUS** sur DISABLED sur chaque gestionnaire de files d'attente du cluster.

Comme décrit dans [«Routage direct dans les clusters de publication / abonnement»](#), à la page 84, lorsque vous introduisez une rubrique de cluster routée directement dans un cluster, tous les référentiels partiels sont automatiquement notifiés de tous les autres membres du cluster. La rubrique en cluster peut également créer des abonnements sur tous les autres noeuds (par exemple, où **PROXYSUB (FORCE)**

est spécifié) et entraîner le démarrage d'un grand nombre de canaux à partir d'un gestionnaire de files d'attente, même en l'absence d'abonnements locaux. Cela permet de placer une charge supplémentaire immédiate sur chaque gestionnaire de files d'attente du cluster. Pour un cluster contenant de nombreux gestionnaires de files d'attente, cela peut entraîner une réduction significative des performances. Par conséquent, l'introduction de la publication / l'abonnement routé directement à un cluster doit être soigneusement planifiée.

Lorsque vous savez qu'un cluster ne peut pas prendre en charge les surcharges de la publication / l'abonnement routé directement, vous pouvez utiliser à la place l'hôte de rubrique routé par la publication / l'abonnement. Pour une présentation des différences, voir [«Conception des clusters de publication/abonnement»](#), à la page 82.

Si vous préférez désactiver complètement la fonctionnalité de publication / abonnement pour le cluster, vous pouvez le faire en définissant l'attribut de gestionnaire de files d'attente **PSCLUS** sur DISABLED sur chaque gestionnaire de files d'attente du cluster. Ce paramètre désactive la fonction de publication / abonnement de l'hôte de rubrique et de l'hôte de rubrique dans le cluster en modifiant trois aspects de la fonctionnalité du gestionnaire de files d'attente:

- Un administrateur de ce gestionnaire de files d'attente ne peut plus définir un objet Topic en tant que cluster.
- Les définitions de rubrique entrantes ou les abonnements de proxy provenant d'autres gestionnaires de files d'attente sont rejetés et un message d'avertissement est consigné pour informer l'administrateur d'une configuration incorrecte.
- Les référentiels complets ne partagent plus automatiquement les informations sur chaque gestionnaire de files d'attente avec tous les autres référentiels partiels lorsqu'ils reçoivent une définition de rubrique.

Bien que **PSCLUS** soit un paramètre de chaque gestionnaire de files d'attente individuel d'un cluster, il n'est pas destiné à désactiver de manière sélective la publication / l'abonnement dans un sous-ensemble de gestionnaires de files d'attente du cluster. Si vous désactivez de cette manière de manière sélective, des messages d'erreur fréquents s'affichent. En effet, les abonnements de proxy et les définitions de rubrique sont constamment vus et rejetés si une rubrique est mise en cluster sur un gestionnaire de files d'attente où **PSCLUS** est activé.

Par conséquent, vous devez définir **PSCLUS** sur DISABLED sur chaque gestionnaire de files d'attente du cluster. Toutefois, dans la pratique, cet état peut être difficile à atteindre et à gérer, par exemple, les gestionnaires de files d'attente peuvent rejoindre et quitter le cluster à tout moment. Au minimum, vous devez vous assurer que **PSCLUS** est défini sur DISABLED sur tous les gestionnaires de files d'attente de référentiel complet. Si vous effectuez cette opération et qu'une rubrique en cluster est ensuite définie sur un gestionnaire de files d'attente ENABLED dans le cluster, les référentiels complets ne sont pas tenus d'informer chaque gestionnaire de files d'attente de tous les autres gestionnaires de files d'attente, de sorte que votre cluster est protégé contre les problèmes de mise à l'échelle potentiels dans tous les gestionnaires de files d'attente. Dans ce scénario, l'origine de la rubrique en cluster est indiquée dans les journaux d'erreurs des gestionnaires de files d'attente de référentiel complet.

Si un gestionnaire de files d'attente participe à un ou plusieurs clusters de publication / abonnement, ainsi qu'à un ou plusieurs clusters point à point, vous devez définir **PSCLUS** sur ENABLED sur ce gestionnaire de files d'attente. Pour cette raison, lorsque vous chevauchez un cluster point à point avec un cluster de publication / abonnement, vous devez utiliser un ensemble distinct de référentiels complets dans chaque cluster. Cette approche permet aux définitions de rubrique et aux informations sur chaque gestionnaire de files d'attente de circuler uniquement dans le cluster de publication / abonnement.

Pour éviter des configurations incohérentes lorsque vous remplacez **PSCLUS** par ENABLED par DISABLED, aucun objet de rubrique en cluster ne peut exister dans un cluster dont ce gestionnaire de files d'attente est membre. Toutes les rubriques de ce type, même celles qui sont définies à distance, doivent être supprimées avant de remplacer **PSCLUS** par DISABLED.

Pour plus d'informations sur **PSCLUS**, voir [ALTER QMGR \(PSCLUS\)](#).

### Concepts associés

[Performances du cluster de publication / abonnement par routage direct](#)

## Publication / abonnement et clusters multiples

Un seul gestionnaire de files d'attente peut être membre de plusieurs clusters. Cette disposition est parfois appelée *chevauchement de clusters*. Grâce à ce chevauchement, les files d'attente en cluster peuvent être rendues accessibles à partir de plusieurs clusters et le trafic de messages point à point peut être acheminé des gestionnaires de files d'attente d'un cluster vers les gestionnaires de files d'attente d'un autre cluster. Les rubriques en cluster dans les clusters de publication / abonnement ne fournissent pas la même fonctionnalité. Par conséquent, leur comportement doit être clairement compris lors de l'utilisation de plusieurs clusters.

Contrairement à une file d'attente, vous ne pouvez pas associer une définition de rubrique à plusieurs clusters. La portée d'une rubrique en cluster est limitée à ces gestionnaires de files d'attente dans le même cluster que celui pour lequel la rubrique est définie. Cela permet de propager les publications aux abonnements uniquement sur les gestionnaires de files d'attente du même cluster.

## Arborescence de rubriques d'un gestionnaire de files d'attente

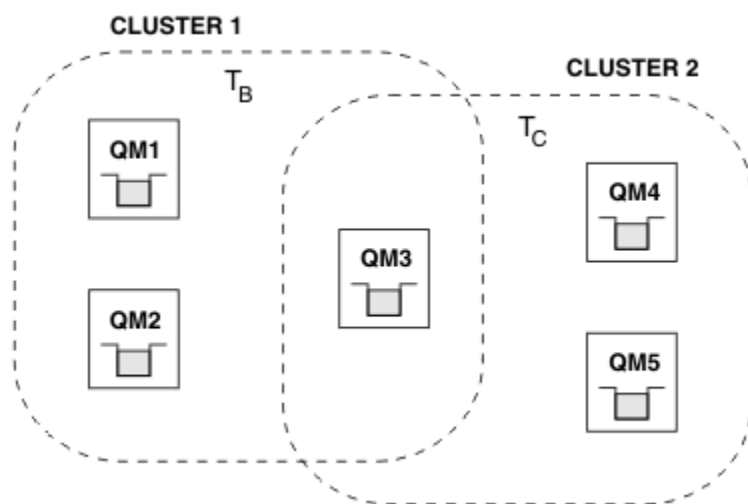


Figure 28. Clusters qui se chevauchent: deux clusters s'abonnant chacun à des rubriques différentes

Lorsqu'un gestionnaire de files d'attente est membre de plusieurs clusters, il est informé de toutes les rubriques en cluster définies dans chacun de ces clusters. Par exemple, dans la figure précédente, QM3 reconnaît les objets de rubrique  $T_B$  et  $T_C$  administrés en cluster, alors que QM1 ne connaît que  $T_B$ . QM3 applique les deux définitions de rubrique à sa rubrique locale et a donc un comportement différent de QM1 pour certaines rubriques. Pour cette raison, il est important que les rubriques en cluster provenant de différents clusters n'interfèrent pas les unes avec les autres. Des interférences peuvent se produire lorsqu'une rubrique en cluster est définie au-dessus ou en dessous d'une autre rubrique en cluster dans un cluster différent (par exemple, ils ont des chaînes de rubrique /Sport et /Sport/Football) ou même pour la même chaîne de rubrique dans les deux. Une autre forme d'interférence est lorsque des objets de rubrique en cluster gérés sont définis avec le même nom d'objet dans des clusters différents, mais pour des chaînes de rubrique différentes.

Si une telle configuration est effectuée, la distribution des publications aux abonnements correspondants devient très dépendante des emplacements relatifs des éditeurs et des abonnés par rapport au cluster. Pour cette raison, vous ne pouvez pas vous fier à une telle configuration et vous devez la modifier pour supprimer les rubriques interférentes.

Lors de la planification d'une topologie de cluster qui se chevauche avec la messagerie de publication / abonnement, vous pouvez éviter toute interférence en traitant l'arborescence de rubriques et les noms d'objet de rubrique en cluster comme s'ils s'étendent sur tous les clusters qui se chevauchent dans la topologie.



## Intégration de plusieurs clusters de publication / abonnement

S'il est nécessaire que la messagerie de publication / abonnement couvre les gestionnaires de files d'attente dans des clusters différents, deux options sont disponibles:

- Connectez les clusters ensemble à l'aide d'une configuration de hiérarchie de publication / abonnement. Voir [Combinaison d'espaces de sujet de plusieurs clusters](#).
- Créez un cluster supplémentaire qui superpose les clusters existants et inclut tous les gestionnaires de files d'attente qui doivent publier ou s'abonner à une rubrique particulière.

Avec cette dernière option, vous devez examiner attentivement la taille du cluster et le mécanisme de routage de cluster le plus efficace. Voir «[Conception des clusters de publication/abonnement](#)», à la page 82.

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

Il existe quelques restrictions à prendre en compte lors de la conception d'un cluster de publication / abonnement pour utiliser les publications conservées.

### **Considérations**

*Remarque 1:* Les gestionnaires de files d'attente de cluster suivants stockent toujours la dernière version d'une publication conservée:

- Gestionnaire de files d'attente du diffuseur de publications
- Dans un cluster routé par un hôte de rubrique, l'hôte de rubrique (à condition qu'il n'y ait qu'un seul hôte de rubrique pour la rubrique, comme expliqué dans la section suivante de cet article)
- Tous les gestionnaires de files d'attente dont les abonnements correspondent à la chaîne de rubrique de la publication conservée

*Remarque 2:* les gestionnaires de files d'attente ne reçoivent pas les publications conservées mises à jour alors qu'ils n'ont pas d'abonnement. Par conséquent, toute publication conservée stockée sur un gestionnaire de files d'attente qui ne s'abonne plus à la rubrique devient périmée.

*Remarque 3:* lors de la création d'un abonnement, s'il existe une copie locale d'une publication conservée pour la chaîne de rubrique, la copie locale est distribuée à l'abonnement. Si vous êtes le premier abonné pour une chaîne de rubrique donnée, une publication conservée correspondante est également distribuée à partir de l'un des membres de cluster suivants:

- Dans un cluster routé directement, le gestionnaire de files d'attente du diffuseur de publications
- Dans un cluster routé par un hôte de rubrique, les hôtes de rubrique pour la rubrique donnée

La distribution d'une publication conservée d'un hôte de rubrique ou d'un gestionnaire de files d'attente de publication au gestionnaire de files d'attente d'abonnement est asynchrone par rapport aux appels `MQSUB`. Par conséquent, si vous utilisez l'appel `MQSUBRQ`, la dernière publication conservée peut être omise jusqu'à un appel ultérieur à `MQSUBRQ`.

### **Implications**

Pour tout cluster de publication / abonnement, lorsqu'un premier abonnement est effectué, le gestionnaire de files d'attente local peut stocker une copie périmée d'une publication conservée et il s'agit de la copie qui est distribuée au nouvel abonnement. L'existence d'un abonnement sur le gestionnaire de files d'attente local signifie que cela sera résolu lors de la prochaine mise à jour de la publication conservée.

Pour un cluster de publication / abonnement routé par un hôte de rubrique, si vous configurez plusieurs hôtes de rubrique pour une rubrique donnée, les nouveaux abonnés peuvent recevoir la dernière publication conservée d'un hôte de rubrique ou ils peuvent recevoir une publication conservée périmée d'un autre hôte de rubrique (la dernière ayant été perdue). Pour le routage via un hôte de rubrique, il est habituel de configurer plusieurs hôtes de rubrique pour une rubrique donnée. Toutefois, si vous prévoyez

que les applications utilisent des publications conservées, vous ne devez configurer qu'un seul hôte de rubrique pour chaque rubrique.

Pour une chaîne de rubrique donnée, vous ne devez utiliser qu'un seul diffuseur et vous assurer que le diffuseur utilise toujours le même gestionnaire de files d'attente. Si vous ne le faites pas, différentes publications conservées peuvent être actives sur différents gestionnaires de files d'attente pour la même rubrique, ce qui entraîne un comportement inattendu. Etant donné que plusieurs abonnements de proxy sont distribués, plusieurs publications conservées peuvent être reçues.

Si vous êtes toujours préoccupé par les abonnés qui utilisent des publications périmées, envisagez de définir un délai d'expiration de message lorsque vous créez chaque publication conservée.

Vous pouvez utiliser la commande **CLEAR TOPICSTR** pour supprimer une publication conservée d'un cluster de publication / abonnement. Dans certaines circonstances, vous devrez peut-être exécuter la commande sur plusieurs membres du cluster de publication / abonnement, comme décrit dans **CLEAR TOPICSTR**.

## Abonnements génériques et publications conservées

Si vous utilisez des abonnements génériques, les abonnements de proxy correspondants distribués aux autres membres du cluster de publication / abonnement sont des caractères génériques du séparateur de rubrique immédiatement avant le premier caractère générique. Voir Rubriques relatives aux caractères génériques et aux clusters.

Par conséquent, le caractère générique utilisé peut correspondre à un plus grand nombre de chaînes de rubrique et de publications conservées que l'application d'abonnement.

Cela augmente la quantité de stockage nécessaire pour les publications conservées et vous devez donc vous assurer que les gestionnaires de files d'attente d'hébergement disposent d'une capacité de stockage suffisante.

### Concepts associés

Publications conservées

Transfert d'abonnement de proxy individuel et publication partout

## ***Considérations relatives à l'ACTUALISATION DU CLUSTER pour les clusters de publication / abonnement***

L'émission de la commande **REFRESH CLUSTER** entraîne la suppression temporaire par le gestionnaire de files d'attente des informations locales sur un cluster, y compris les rubriques de cluster et les abonnements de proxy associés.

Le temps écoulé entre l'émission de la commande **REFRESH CLUSTER** et le moment où le gestionnaire de files d'attente récupère une connaissance complète des informations nécessaires pour la publication / l'abonnement en cluster dépend de la taille du cluster, de la disponibilité et de la réactivité des gestionnaires de files d'attente de référentiel complet.

Lors du processus d'actualisation, une interruption du trafic de publication / abonnement dans un cluster de publication / abonnement se produit. Pour les clusters de grande taille, l'utilisation de la commande **REFRESH CLUSTER** peut interrompre le cluster pendant qu'il est en cours, et à nouveau à des intervalles de 27 jours par la suite lorsque les objets de cluster envoient automatiquement des mises à jour de statut à tous les gestionnaires de files d'attente intéressés. Voir L'actualisation d'un grand cluster peut affecter les performances et la disponibilité du cluster. Pour ces raisons, la commande **REFRESH CLUSTER** doit être utilisée dans un cluster de publication / abonnement uniquement sous la supervision de votre centre de support IBM.

L'interruption du cluster peut apparaître en externe comme les symptômes suivants:

- Les abonnements aux rubriques de cluster de ce gestionnaire de files d'attente ne reçoivent pas les publications des diffuseurs de publications qui sont connectés à d'autres gestionnaires de files d'attente du cluster.
- Les messages publiés dans les rubriques de cluster de ce gestionnaire de files d'attente ne sont pas propagés aux abonnements d'autres gestionnaires de files d'attente.

- Les abonnements aux rubriques de cluster de ce gestionnaire de files d'attente créés au cours de cette période ne permettent pas d'envoyer de manière cohérente des abonnements de proxy à d'autres membres du cluster.
- Les abonnements aux rubriques de cluster de ce gestionnaire de files d'attente supprimés au cours de cette période ne suppriment pas systématiquement les abonnements de proxy des autres membres du cluster.
- Pausages de 10 secondes ou plus dans la distribution des messages.
- **MQPUT** , par exemple, **MQRC\_PUBLICATION\_FAILURE**.
- Publications placées dans la file d'attente de rebut pour la raison **MQRC\_UNKNOWN\_REMOTE\_Q\_MGR**

Pour ces raisons, les applications de publication / abonnement doivent être mises au repos avant d'émettre la commande **REFRESH CLUSTER** .

Une fois qu'une commande **REFRESH CLUSTER** a été émise sur un gestionnaire de files d'attente dans un cluster de publication / abonnement, attendez que tous les gestionnaires de files d'attente de cluster et toutes les rubriques de cluster aient été actualisés, puis resynchronisez les abonnements de proxy comme décrit dans [Resynchronisation des abonnements de proxy](#). Une fois que tous les abonnements de proxy ont été correctement resynchronisés, redémarrez vos applications de publication / abonnement.

Si l'exécution d'une commande **REFRESH CLUSTER** prend beaucoup de temps, surveillez-la en examinant le **CURDEPTH** de **SYSTEM.CLUSTER.COMMAND.QUEUE**.

### Concepts associés

«[Mise en cluster : meilleures pratiques d'utilisation REFRESH CLUSTER](#)», à la page 75

Vous utilisez la commande **REFRESH CLUSTER** pour supprimer toutes les informations stockées localement sur un cluster et régénérer ces informations à partir des référentiels complets du cluster. Vous ne devez pas utiliser cette commande, sauf dans des circonstances exceptionnelles. Si vous avez besoin de l'utiliser, il y a des considérations spéciales sur la façon dont vous l'utilisez. Ces informations sont un guide basé sur les tests et les commentaires des clients.

### Référence associée

[Problèmes d'application rencontrés lors de l'exécution de REFRESH CLUSTER](#)

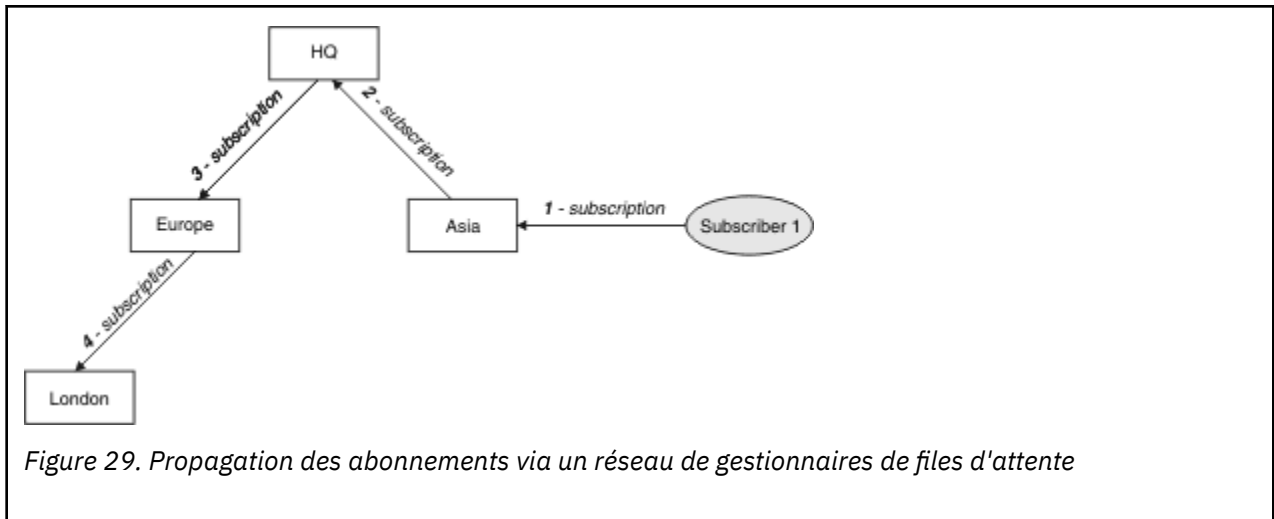
[Référence des commandes MQSC: REFRESH CLUSTER](#)

## Routage dans les hiérarchies de publication / abonnement

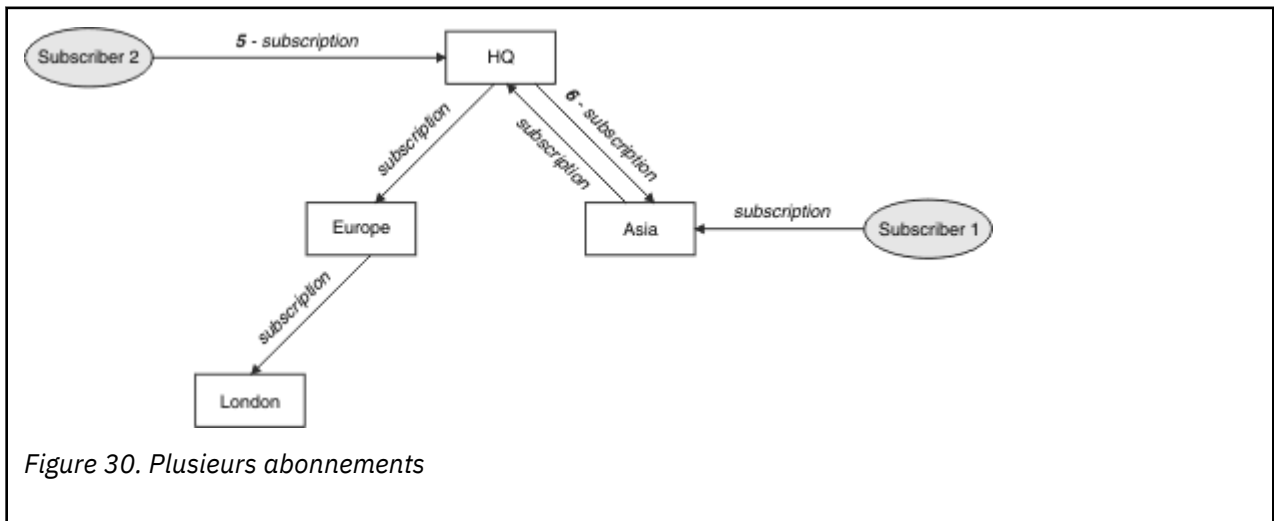
Si votre topologie de gestionnaire de files d'attente réparties est une hiérarchie de publication / abonnement et qu'un abonnement est effectué sur un gestionnaire de files d'attente, un abonnement proxy est créé par défaut sur chaque gestionnaire de files d'attente de la hiérarchie. Les publications reçues sur un gestionnaire de files d'attente sont ensuite acheminées via la hiérarchie vers chaque gestionnaire de files d'attente qui héberge un abonnement correspondant.

Pour une présentation de la façon dont les messages sont acheminés entre les gestionnaires de files d'attente dans les hiérarchies de publication / abonnement et les clusters, voir [Réseaux de publication / abonnement distribués](#).

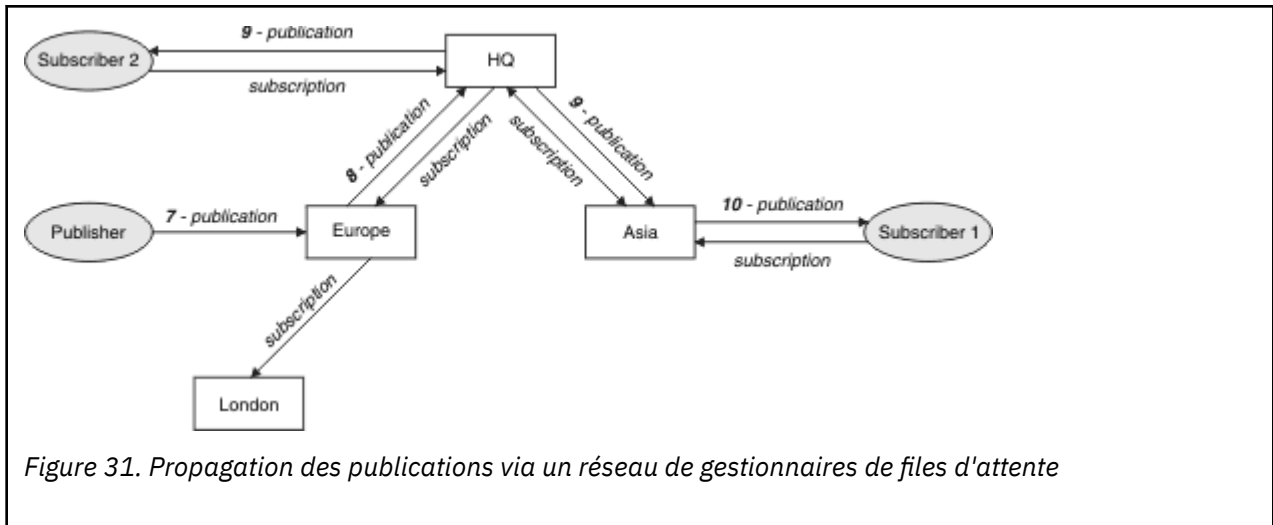
Lorsqu'un abonnement à une rubrique est effectué sur un gestionnaire de files d'attente dans une hiérarchie de publication / abonnement distribuée, le gestionnaire de files d'attente gère le processus par lequel l'abonnement est propagé aux gestionnaires de files d'attente connectés. Les *abonnements de proxy* sont transmis à tous les gestionnaires de files d'attente du réseau. Un abonnement de proxy fournit à un gestionnaire de files d'attente les informations dont il a besoin pour transmettre une publication aux gestionnaires de files d'attente qui hébergent des abonnements pour cette rubrique. Chaque gestionnaire de files d'attente d'une hiérarchie de publication / abonnement ne connaît que ses relations directes. Les publications placées dans un gestionnaire de files d'attente sont envoyées, via ses relations directes, aux gestionnaires de files d'attente ayant des abonnements. Ceci est illustré dans la figure suivante, dans laquelle *Abonné 1* enregistre un abonnement pour une rubrique particulière sur le gestionnaire de files d'attente *Asie* (1). Les abonnements de proxy pour cet abonnement sur le gestionnaire de files d'attente *Asie* sont réacheminés vers tous les autres gestionnaires de files d'attente du réseau (2,3, 4).



Un gestionnaire de files d'attente consolide tous les abonnements qui y sont créés, que ce soit à partir d'applications locales ou de gestionnaires de files d'attente éloignées. Il crée des abonnements de proxy pour les rubriques des abonnements avec ses voisins, sauf s'il existe déjà un abonnement de proxy. Ceci est illustré dans la figure suivante, dans laquelle *Subscriber 2* enregistre un abonnement, à la même rubrique que dans Figure 29, à la page 116, sur le gestionnaire de files d'attente *HQ* (5). L'abonnement à cette rubrique est réacheminé vers le gestionnaire de files d'attente *Asie* afin qu'il soit conscient que des abonnements existent ailleurs sur le réseau (6). L'abonnement n'est pas réacheminé vers le gestionnaire de files d'attente *Europe* car un abonnement a déjà été enregistré pour cette rubrique ; voir l'étape 3 dans Figure 29, à la page 116.



Lorsqu'une application publie des informations dans une rubrique, le gestionnaire de files d'attente de réception les transmet par défaut à tous les gestionnaires de files d'attente ayant des abonnements valides à la rubrique. Il peut être réacheminé via un ou plusieurs gestionnaires de files d'attente intermédiaires. Ceci est illustré dans la figure suivante, dans laquelle un diffuseur de publications envoie une publication, sur la même rubrique que dans Figure 30, à la page 116, au gestionnaire de files d'attente *Europe* (7). Un abonnement à cette rubrique existe depuis *HQ* vers *Europe*, de sorte que la publication est transmise au gestionnaire de files d'attente *HQ* (8). Toutefois, aucun abonnement n'existe depuis *Londres* vers *Europe* (uniquement depuis *Europe* vers *Londres*), de sorte que la publication n'est pas réacheminée vers le gestionnaire de files d'attente *Londres*. Le gestionnaire de files d'attente *HQ* envoie la publication directement à *Subscriber 2* et au gestionnaire de files d'attente *Asie* (9). La publication est transmise à l'Abonné 1 à partir de *Asie* (10).



Lorsqu'un gestionnaire de files d'attente envoie des publications ou des abonnements à un autre gestionnaire de files d'attente, il définit son propre ID utilisateur dans le message. Si vous utilisez une hiérarchie de publication / abonnement et si le canal entrant est configuré pour insérer des messages avec les droits de l'ID utilisateur dans le message, vous devez autoriser l'ID utilisateur du gestionnaire de files d'attente d'envoi. Voir [Utilisation des ID utilisateur par défaut avec une hiérarchie de gestionnaires de files d'attente](#).

**Remarque :** Si vous utilisez à la place des clusters de publication / abonnement, l'autorisation est gérée par le cluster.

## Résumé et remarques supplémentaires

Une hiérarchie de publication / abonnement vous donne un contrôle précis sur la relation entre les gestionnaires de files d'attente. Après avoir été créé, il a besoin de peu d'intervention manuelle pour l'administrer. Cependant, il impose également certaines contraintes à votre système:

- Les noeuds supérieurs de la hiérarchie, en particulier le noeud racine, doivent être hébergés sur des équipements robustes, hautement disponibles et performants. Cela est dû au fait que le trafic des publications devrait passer par ces noeuds.
- La disponibilité de chaque gestionnaire de files d'attente non-feuille dans la hiérarchie affecte la capacité du réseau à transmettre des messages des diffuseurs aux abonnés des autres gestionnaires de files d'attente.
- Par défaut, toutes les chaînes de rubrique souscrites sont propagées dans toute la hiérarchie et les publications sont propagées uniquement aux gestionnaires de files d'attente éloignées ayant un abonnement à la rubrique associée. Par conséquent, des changements rapides de l'ensemble des abonnements peuvent devenir un facteur limitatif. Vous pouvez modifier ce comportement par défaut et faire en sorte que toutes les publications soient propagées à tous les gestionnaires de files d'attente, ce qui élimine le besoin d'abonnements de proxy. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

**Remarque :** Une restriction similaire s'applique également aux clusters routés directement.

- En raison de la nature interconnectée des gestionnaires de files d'attente de publication / abonnement, la propagation des abonnements proxy sur tous les noeuds du réseau prend du temps. Les publications distantes ne commencent pas nécessairement à être abonnées immédiatement, de sorte que les publications précoces peuvent ne pas être envoyées à la suite d'un abonnement à une nouvelle chaîne de rubrique. Vous pouvez supprimer les problèmes causés par le délai d'abonnement en propageant toutes les publications à tous les gestionnaires de files d'attente, ce qui élimine le besoin d'abonnements de proxy. Voir [Performances des abonnements dans les réseaux de publication/abonnement](#).

**Remarque :** Cette restriction s'applique également aux clusters routés directement.

- Pour une hiérarchie de publication / abonnement, l'ajout ou la suppression de gestionnaires de files d'attente nécessite une configuration manuelle de la hiérarchie, en tenant compte de l'emplacement de ces gestionnaires de files d'attente et de leur dépendance vis-à-vis d'autres gestionnaires de files d'attente. A moins que vous n'ajoutiez ou ne supprimiez des gestionnaires de files d'attente qui se trouvent en bas de la hiérarchie et que vous n'ayez donc pas d'autres branches en dessous, vous devrez également configurer d'autres gestionnaires de files d'attente dans la hiérarchie.

Avant d'utiliser une hiérarchie de publication / abonnement comme mécanisme de routage, explorez les approches alternatives détaillées dans «Routage direct dans les clusters de publication / abonnement», à la page 84 et «Routage de l'hôte de rubrique dans les clusters de publication / abonnement», à la page 89.

## Files d'attente du système de publication / abonnement distribué

Quatre files d'attente système sont utilisées par les gestionnaires de files d'attente pour la messagerie de publication / abonnement. Vous devez être conscient de leur existence uniquement à des fins d'identification des problèmes et de planification de la capacité.

Voir [Equilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement](#) pour savoir comment surveiller ces files d'attente.



File d'attente système	Objet
SYSTEM.INTER.QMGR.CONTROL	File d'attente de contrôle de publication / abonnement répartie IBM MQ
SYSTEM.INTER.QMGR.FANREQ	File d'attente d'entrée du processus de distribution de l'abonnement au proxy interne de publication / abonnement distribué IBM MQ
SYSTEM.INTER.QMGR.PUBS	Publications de publication / abonnement distribué IBM MQ
SYSTEM.HIERARCHY.STATE	Etat de la relation de hiérarchie de publication / abonnement distribué IBM MQ

**z/OS** Sous z/OS, vous configurez les objets système nécessaires lorsque vous créez le gestionnaire de files d'attente, en incluant les exemples CSQ4INSX, CSQ4INSR et CSQ4INSG dans le fichier d'entrée d'initialisation CSQINP2 . Pour plus d'informations, voir [Tâche 13: Personnalisation des fichiers d'entrée d'initialisation](#).

Les attributs des files d'attente système de publication / abonnement sont affichés dans [Tableau 7](#), à la page 118.

Attribut	Valeur par défaut
DEFPSIST	Oui
DEFSOPT	PARTAGÉ
MAXMSGL	<p><b>Multi</b> Sous Multiplateformes: valeur du paramètre MAXMSGL de la commande ALTER QMGR</p> <p><b>z/OS</b> Sous z/OS: 4194304 (c'est-à-dire 4 Mo)</p>
MAXDEPTH	999999999
PARTAGER	Non disponible

Tableau 7. Attributs des files d'attente système de publication / abonnement (suite)

Attribut	Valeur par défaut
  STGCLASS	Cet attribut est utilisé uniquement sur les plateformes z/OS

**Remarque :** La seule file d'attente contenant les messages insérés par les applications est SYSTEM. INTER. QMGR. PUBS. **MAXDEPTH** est défini sur sa valeur maximale pour cette file d'attente afin de permettre la génération temporaire de messages publiés pendant les indisponibilités ou les périodes de charge excessive. Si le gestionnaire de files d'attente s'exécute sur un système sur lequel cette longueur de file d'attente n'a pas pu être contenue, vous devez l'ajuster.

#### Tâches associées

Identification et résolution des problèmes de publication / abonnement distribué

#### **Erreurs de file d'attente du système de publication / abonnement distribué**

Des erreurs peuvent se produire lorsque les files d'attente du gestionnaire de files d'attente de publication / abonnement réparties ne sont pas disponibles. Cela affecte la propagation des connaissances d'abonnement sur le réseau de publication / abonnement et la publication vers les abonnements sur les gestionnaires de files d'attente éloignées.

Si la file d'attente des demandes de sortance SYSTEM. INTER. QMGR. FANREQ n'est pas disponible, la création d'un abonnement peut générer une erreur et des messages d'erreur sont écrits dans le journal des erreurs du gestionnaire de files d'attente lorsque des abonnements de proxy doivent être distribués aux gestionnaires de files d'attente directement connectés.

Si la file d'attente d'état de la relation de hiérarchie SYSTEM. HIERARCHY. STATE n'est pas disponible, un message d'erreur est consigné dans le journal des erreurs du gestionnaire de files d'attente et le moteur de publication / abonnement passe en mode COMPAT. Pour afficher le mode publication / abonnement, utilisez la commande DISPLAY QMGR PSMODE.

Si d'autres files d'attente SYSTEM. INTER. QMGR ne sont pas disponibles, un message d'erreur est consigné dans le journal des erreurs du gestionnaire de files d'attente et, bien que la fonction ne soit pas désactivée, il est probable que les messages de publication / abonnement s'accumulent dans les files d'attente sur ce gestionnaire de files d'attente ou sur les gestionnaires de files d'attente éloignées.

Si la file d'attente système de publication / abonnement ou la file d'attente de transmission requise vers un gestionnaire de files d'attente de cluster parent, enfant ou de publication / abonnement n'est pas disponible, les résultats suivants se produisent:

- Les publications ne sont pas distribuées et une application de publication peut recevoir une erreur. Pour plus de détails sur le moment où l'application de publication reçoit une erreur, voir les paramètres suivants de la commande **DEFINE TOPIC : PMSGDLV**, **NMSGDLV** et **USEDLQ**.
- Les publications inter-gestionnaires de files d'attente reçues sont annulées dans la file d'attente d'entrée, puis font l'objet d'une nouvelle tentative. Si le seuil d'annulation est atteint, les publications non distribuées sont placées dans la file d'attente des messages non livrés. Le journal des erreurs du gestionnaire de files d'attente contiendra des détails sur le problème.
- Un abonnement proxy non distribué est annulé dans la file d'attente des demandes de sortance, puis fait l'objet d'une nouvelle tentative. Si le seuil d'annulation est atteint, l'abonnement proxy non distribué n'est pas distribué à un gestionnaire de files d'attente connecté et est placé dans la file d'attente de rebut. Le journal des erreurs du gestionnaire de files d'attente contient les détails du problème, y compris les détails de toute action administrative corrective nécessaire.
- Les messages de protocole de relation de hiérarchie échouent et le statut de connexion est marqué comme ERROR. Pour afficher le statut de la connexion, utilisez la commande **DISPLAY PUBSUB**.

#### Tâches associées

Identification et résolution des problèmes de publication / abonnement distribué

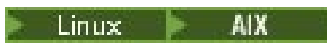


## Multi **Planification de vos exigences de stockage et de performances sur Multiplatforms**

Vous devez définir un stockage réaliste et réalisable, ainsi que des objectifs de performances pour votre système IBM MQ . Utilisez les liens pour découvrir les facteurs qui affectent le stockage et les performances sur votre plateforme.




Les exigences varient en fonction des systèmes sur lesquels vous utilisez IBM MQ et des composants que vous souhaitez utiliser.

Pour obtenir les informations les plus récentes sur les environnements matériels et logiciels pris en charge, voir [Configuration système requise pour IBM MQ](#).

IBM MQ stocke les données du gestionnaire de files d'attente dans le système de fichiers. Utilisez les liens suivants pour en savoir plus sur la planification et la configuration des structures de répertoire à utiliser avec IBM MQ:

- [«Planification de la prise en charge du système de fichiers sur Multiplatforms»](#), à la page 123
- [«Configuration requise pour les systèmes de fichiers partagés sur Multiplatforms»](#), à la page 124
- [«Partage de fichiers IBM MQ sur Multiplatforms»](#), à la page 133
-  [«Structure de répertoires sur les systèmes AIX and Linux»](#), à la page 136
-  [«Structure de répertoires sur les systèmes Windows»](#), à la page 145
-  [«Structure de répertoire sous IBM i»](#), à la page 148

Utilisez les liens suivants pour obtenir des informations sur les ressources système, la mémoire partagée et la priorité des processus sous AIX and Linux:

-  [«Ressources IPC IBM MQ et UNIX System V»](#), à la page 153
-  [«Mémoire partagée sur AIX»](#), à la page 153
-  [«Priorité des processus IBM MQ et UNIX»](#), à la page 153

Utilisez les liens suivants pour obtenir des informations sur les fichiers journaux:

- [«Choix de la journalisation circulaire ou linéaire sur Multiplatforms»](#), à la page 152
- [Calcul de la taille du journal](#)

### Concepts associés

[«Planning your IBM MQ environment on z/OS»](#), à la page 154

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

### Tâches associées

[«Planification d'une architecture IBM MQ»](#), à la page 5

Lors de la planification de l'environnement IBM MQ, prenez en compte la prise en charge fournie par IBM MQ pour les architectures à un ou plusieurs gestionnaires de files d'attente et pour les styles de messagerie de publication/abonnement et point à point. Prévoyez également vos exigences en matières de ressources et votre utilisation des fonctions de consignation et de sauvegarde.

### Référence associée

[Configuration matérielle et logicielle requise sous AIX and Linux](#)

[Configuration matérielle et logicielle requise sous Windows](#)

## Multi **Espace disque requis sur Multiplatforms**

Les besoins en espace disque pour IBM MQ dépendent des composants que vous installez et de l'espace disque dont vous avez besoin pour votre travail.



Vous avez besoin d'espace de stockage sur disque pour les composants facultatifs que vous avez choisi d'installer, notamment les composants prérequis. L'espace disque total requis varie également en fonction du nombre de files d'attente utilisées, du nombre et de la taille des messages dans les files et du type des messages (persistant ou non). Vous devez pouvoir archiver sur disque, bande ou sur tout autre support et vous avez besoin d'espace disponible pour vos propres programmes d'applications.

Les tableaux suivants montrent l'espace disque approximatif requis lorsque vous installez différentes combinaisons du produit sur différentes plateformes. (Les valeurs sont arrondies à la valeur la plus proche de 5 Mo, sachant qu'1 Mo correspond à 1 048 576 octets.)

- **LTS** «Espace disque requis pour Long Term Support», à la page 121
- **CD** «Espace disque requis pour Continuous Delivery», à la page 122

## Espace disque requis pour Long Term Support

V 9.4.0 LTS

Tableau 8. IBM MQ Espace disque requis pour Multiplatforms pour Long Term Support

Plateforme	installation du client <sup>«1»</sup> , à la page 121	Installation du serveur <sup>«2»</sup> , à la page 121	Installation complète <sup>«3»</sup> , à la page 121
<b>AIX</b> AIX	335 Mo	375 Mo	1810 Mo
<b>IBM i</b> IBM i (voir Notes supplémentaires pour IBM i)	485 Mo	845 Mo	1965 Mo
<b>Linux</b> Linux for x86-64	270 Mo	295 Mo	2010 Mo
<b>Linux</b> Linux on Power Systems - Little Endian	170 Mo	190 Mo	1400 Mo
<b>Linux</b> Linux for IBM Z	255 Mo	290 Mo	1485 Mo
<b>Windows</b> Windows (installation 64 bits) <sup>«4»</sup> , à la page 121	295 Mo	425 Mo	2310 Mo

### Remarques :











1. Une installation client comprend les composants suivants :
  - Environnement d'exécution
  - Environnement
2. Une installation serveur comprend les composants suivants :
  - Environnement d'exécution
  - serveur
3. Une installation complète comprend tous les composants disponibles.
4. **Windows** Tous les composants répertoriés ici ne sont pas des fonctions installables sur les systèmes Windows ; leurs fonctionnalités sont parfois incluses dans d'autres fonctions. Voir [Fonctions d'IBM MQ pour les systèmes Windows](#).

## Remarques supplémentaires pour IBM i :

1. Sous IBM i, vous ne pouvez pas séparer le client natif du serveur. La figure du serveur dans le tableau correspond à 5724H72\*BASE sans Java, avec l'option English Language Load (2924). Il existe 22 charges linguistiques uniques possibles.
2. La figure de la table est destinée au client natif 5725A49 \*BASE sans Java.
3. Les classes Java et JMS peuvent être ajoutées aux liaisons serveur et client. Si vous souhaitez inclure ces fonctions, ajoutez 110 Mo.
4. L'ajout d'exemples de source au client ou au serveur ajoute 10 Mo supplémentaires.
5. L'ajout d'exemples à des classes Java et JMS ajoute 5 Mo supplémentaires.

## Espace disque requis pour Continuous Delivery

Edition Platform/CD	installation du client <sup>«1»</sup> , à la page 122	Installation du serveur <sup>«2»</sup> , à la page 122	Installation complète <sup>«3»</sup> , à la page 122
 <b>AIX</b>			
 IBM MQ 9.4.0	355 Mo	390 Mo	1440 Mo
 <b>Linux for x86-64 (64 bits)</b>			
 IBM MQ 9.4.0	280 Mo	295 Mo	1195 Mo
 <b>Linux on Power Systems - Little Endian</b>			
 IBM MQ 9.4.0	170 Mo	195 Mo	1075 Mo
 <b>Linux pour IBM Z</b>			
 IBM MQ 9.4.0	260 Mo	290 Mo	1160 Mo
 <b>Windows (installation 64 bits)</b> <sup>«4»</sup> , à la page 123			
 IBM MQ 9.4.0	300 Mo	425 Mo	1785 Mo

### Remarques :

1. Une installation client comprend les composants suivants :
  - Environnement d'exécution
  - Environnement
2. Une installation serveur comprend les composants suivants :
  - Environnement d'exécution
  - serveur
3. Une installation complète comprend tous les composants disponibles.

4. **Windows** Tous les composants répertoriés ici ne sont pas des fonctions installables sur les systèmes Windows ; leurs fonctionnalités sont parfois incluses dans d'autres fonctions. Voir [Fonctions d'IBM MQ pour les systèmes Windows](#).

### Concepts associés

[Composants et fonctions d'IBM MQ](#)

## Multi **Planification de la prise en charge du système de fichiers sur Multiplatforms**

Les données du gestionnaire de files d'attente sont stockées dans le système de fichiers. Un gestionnaire de files d'attente utilise le verrouillage du système de fichiers pour empêcher que plusieurs instances d'un gestionnaire de files d'attente multi-instance soient actives en même temps.

### Systèmes de fichiers partagés

Les systèmes de fichiers partagés permettent à plusieurs systèmes d'accéder simultanément à la même unité de stockage physique. Une altération se produit si plusieurs systèmes accèdent directement à la même unité de stockage physique sans aucun moyen d'imposer le verrouillage et le contrôle des accès concurrents. Les systèmes d'exploitation fournissent des systèmes de fichiers locaux avec verrouillage et contrôle des accès concurrents pour les processus locaux ; les systèmes de fichiers réseau fournissent le verrouillage et le contrôle des accès concurrents pour les systèmes distribués.

Historiquement, les systèmes de fichiers en réseau n'ont pas été suffisamment rapides ou ont fourni un verrouillage et un contrôle des accès concurrents suffisants pour répondre aux exigences de consignation des messages. Aujourd'hui, les systèmes de fichiers en réseau peuvent fournir de bonnes performances et les implémentations de protocoles de système de fichiers réseau fiables, tels que le *protocole RFC 3530, Network File System (NFS) version 4*, répondent aux exigences de journalisation des messages de manière fiable.

### Systèmes de fichiers partagés et IBM MQ

Les données de gestionnaire de files d'attente pour un gestionnaire de files d'attente multi-instance sont stockées dans un système de fichiers réseau partagé. Sur les systèmes AIX, Linux, and Windows , les fichiers de données et les fichiers journaux du gestionnaire de files d'attente doivent être placés dans un système de fichiers réseau partagé. **IBM i** Sur IBM i, les journaux sont utilisés à la place des fichiers journaux et les journaux ne peuvent pas être partagés. Les gestionnaires de files d'attente multi-instance sous IBM i utilisent la réplication de journal ou les journaux commutables pour rendre les journaux disponibles entre les différentes instances de gestionnaire de files d'attente.

IBM MQ utilise le verrouillage pour empêcher que plusieurs instances du même gestionnaire de files d'attente multi-instance soient actives simultanément. Le même verrouillage garantit également que deux gestionnaires de files d'attente distincts ne peuvent pas utiliser par inadvertance le même ensemble de fichiers de données de gestionnaire de files d'attente. Une seule instance d'un gestionnaire de files d'attente peut avoir son verrou à la fois. Par conséquent, IBM MQ prend en charge les données de gestionnaire de files d'attente stockées sur le stockage en réseau accessible en tant que système de fichiers partagé.

Etant donné que tous les protocoles de verrouillage des systèmes de fichiers réseau ne sont pas robustes et qu'un système de fichiers peut être configuré pour les performances plutôt que pour l'intégrité des données, vous devez exécuter la commande **amqmfscck** pour tester si un système de fichiers réseau contrôle correctement l'accès aux données et aux journaux du gestionnaire de files d'attente. Cette commande s'applique uniquement aux systèmes UNIX, Linux et IBM i . Sous Windows, il n'existe qu'un seul système de fichiers réseau pris en charge et la commande **amqmfscck** n'est pas requise.

### Tâches associées

«Vérification du comportement du système de fichiers partagé sur Multiplatforms», à la page 126

Exécutez **amqmfscck** pour vérifier si un système de fichiers partagé sous AIX, Linux ou IBM i répond aux exigences de stockage des données de gestionnaire de files d'attente d'un gestionnaire de files d'attente

multi-instance. (La seule condition requise pour une configuration Windows est qu'elle utilise SMB 3 pour la mise à disposition de stockage partagé.)

## **Multi** Configuration requise pour les systèmes de fichiers partagés sur Multiplatforms

Les systèmes de fichiers partagés doivent assurer l'intégrité de l'écriture des données, garantir un accès exclusif aux fichiers et libérer les verrous en cas d'échec de fonctionnement fiable avec IBM MQ.

### Conditions qu'un système de fichiers partagé doit remplir

Il existe trois exigences fondamentales auxquelles un système de fichiers partagé doit satisfaire pour fonctionner de manière fiable avec IBM MQ:

#### 1. Intégrité d'écriture des données

L'intégrité de l'écriture de données est parfois appelée *Ecriture sur le disque lors du vidage*. Le gestionnaire de files d'attente doit pouvoir se synchroniser avec les données validées sur l'unité physique. Dans un système transactionnel, vous devez vous assurer que certaines écritures ont été validées en toute sécurité avant de poursuivre un autre traitement.

Plus spécifiquement, les plateformes IBM MQ for AIX or Linux utilisent l'option d'ouverture `O_SYNC` et l'appel système `fsync()` pour forcer explicitement les écritures sur les supports récupérables, et l'opération d'écriture dépend du bon fonctionnement de ces options.



**Avertissement :** Linux Vous devez monter le système de fichiers avec l'option `async`, qui prend toujours en charge l'option des écritures synchrones et qui offre de meilleures performances que l'option `sync`.

Notez toutefois que si le système de fichiers a été exporté depuis Linux, vous devez toujours exporter le système de fichiers à l'aide de l'option `sync`.

#### 2. Accès exclusif garanti aux fichiers

Pour synchroniser plusieurs gestionnaires de files d'attente, il doit exister un mécanisme permettant à un gestionnaire de files d'attente d'obtenir un verrou exclusif sur un fichier.

#### 3. Libérer les verrous en cas d'échec

En cas de défaillance d'un gestionnaire de files d'attente ou en cas d'incident de communication avec le système de fichiers, les fichiers verrouillés par le gestionnaire de files d'attente doivent être déverrouillés et rendus disponibles pour d'autres processus sans attendre que le gestionnaire de files d'attente soit reconnecté au système de fichiers.

Un système de fichiers partagé doit répondre à ces exigences pour que IBM MQ puisse fonctionner de manière fiable. Si ce n'est pas le cas, les données et les journaux du gestionnaire de files d'attente sont endommagés lors de l'utilisation du système de fichiers partagé dans une configuration de gestionnaire de files d'attente multi-instance.

Pour les gestionnaires de files d'attente multi-instance sur Microsoft Windows, le stockage en réseau doit être accessible par le protocole SMB (Server Message Block) utilisé par les réseaux Microsoft Windows. Le client SMB (Server Message Block) ne répond pas aux exigences de IBM MQ pour le verrouillage de la sémantique sur les plateformes autres que Microsoft Windows. Par conséquent, les gestionnaires de files d'attente multi-instance s'exécutant sur des plateformes autres que Microsoft Windows ne doivent pas utiliser SMB (Server Message Block) comme système de fichiers partagé.

Pour les gestionnaires de files d'attente multi-instance sur d'autres plateformes prises en charge, le stockage doit être accessible par un protocole de système de fichiers réseau compatible Posix et prenant en charge le verrouillage basé sur un bail. Network File System 4 répond à cette exigence. Les systèmes de fichiers plus anciens, tels que Network File System version 3, qui ne disposent pas d'un mécanisme fiable pour libérer les verrous après un incident, ne doivent pas être utilisés avec les gestionnaires de files d'attente multi-instance.

## Vérifie si le système de fichiers partagé répond aux exigences




Vous devez vérifier si le système de fichiers partagé que vous prévoyez d'utiliser répond à ces exigences. Vous devez également vérifier si le système de fichiers est correctement configuré pour la fiabilité. Les systèmes de fichiers partagés fournissent parfois des options de configuration pour améliorer les performances au détriment de la fiabilité.

Pour plus d'informations, voir [Testing statement for IBM MQ multi-instance queue manager file systems](#).

Dans des circonstances normales, IBM MQ fonctionne correctement avec la mise en cache des attributs et il n'est pas nécessaire de désactiver la mise en cache, par exemple en définissant NOAC sur un montage NFS . La mise en cache des attributs peut entraîner des problèmes lorsque plusieurs clients de système de fichiers sont en conflit pour l'accès en écriture au même fichier sur le serveur de système de fichiers, car les attributs mis en cache utilisés par chaque client peuvent ne pas être les mêmes que ceux du serveur. Les journaux d'erreurs du gestionnaire de files d'attente pour un gestionnaire de files d'attente multi-instance sont des exemples de fichiers accessibles de cette manière. Les journaux d'erreurs du gestionnaire de files d'attente peuvent être écrits à la fois par une instance de gestionnaire de files d'attente active et par une instance de gestionnaire de files d'attente de secours et les attributs de fichier mis en cache peuvent entraîner une augmentation de la taille des journaux d'erreurs par rapport à la valeur prévue, avant que le remplacement des fichiers ne se produise.

Pour vous aider à vérifier le système de fichiers, exécutez la tâche [Vérification du comportement du système de fichiers partagé](#). Cette tâche vérifie si votre système de fichiers partagé répond aux exigences 2 et 3. Vous devez vérifier l'exigence 1 dans la documentation de votre système de fichiers partagé ou en expérimentant la consignation des données sur le disque.

Les pannes de disque peuvent provoquer des erreurs lors de l'écriture sur le disque, ce que IBM MQ signale comme des erreurs de capture de données de premier incident. Vous pouvez exécuter le vérificateur de système de fichiers pour votre système d'exploitation afin de vérifier si le système de fichiers partagé contient des erreurs de disque. Exemple :

-   Sous AIX and Linux , le vérificateur de système de fichiers est appelé fsck.
-  Sur les plateformes Windows , le vérificateur de système de fichiers est appelé CHKDSK ou SCANDISK.

## Sécurité du serveur NFS

### Remarques :

- Vous ne pouvez pas utiliser les options **nosuid** ou **noexec** pour un point de montage utilisé pour stocker le répertoire d'installation IBM MQ . En effet, IBM MQ inclut des programmes exécutables setuid / setgid, qui ne doivent pas être empêchés de s'exécuter correctement.
- Lorsque vous placez des données de gestionnaire de files d'attente uniquement sur un serveur NFS (NFS), vous pouvez utiliser les trois options suivantes avec la commande de montage pour sécuriser le système, sans impact négatif sur l'exécution du gestionnaire de files d'attente:

#### **noexec**

En utilisant cette option, vous arrêtez l'exécution des fichiers binaires sur le système NFS, ce qui empêche un utilisateur distant d'exécuter du code indésirable sur le système.

#### **nosuid**

En utilisant cette option, vous empêchez l'utilisation des bits set-user-identifier et set-group-identifier, ce qui empêche un utilisateur distant d'obtenir des privilèges plus élevés.

#### **nodev**

En utilisant cette option, vous arrêtez l'utilisation ou la définition de caractères et bloquez les unités spéciales, ce qui empêche un utilisateur distant de sortir d'une prison chroot.

## partagé sur Multiplatforms

Exécutez **amqmfsc** pour vérifier si un système de fichiers partagé sous AIX, Linux ou IBM i répond aux exigences de stockage des données de gestionnaire de files d'attente d'un gestionnaire de files d'attente multi-instance. (La seule condition requise pour une configuration Windows est qu'elle utilise SMB 3 pour la mise à disposition de stockage partagé.)

### Avant de commencer

Vous avez besoin d'un serveur avec un stockage en réseau et de deux autres serveurs sur lesquels IBM MQ est installé. Vous devez disposer des droits d'administrateur (root) pour configurer le système de fichiers et être un administrateur IBM MQ pour exécuter **amqmfsc**.

### Pourquoi et quand exécuter cette tâche

Le «Configuration requise pour les systèmes de fichiers partagés sur Multiplatforms», à la page 124 décrit la configuration de système de fichiers requise pour l'utilisation d'un système de fichiers partagé avec des gestionnaires de files d'attente multi-instance. La IBM MQ note technique Testing statement for IBM MQ multi-instance queue manager file systems répertorie les systèmes de fichiers partagés avec lesquels IBM a déjà été testé. La procédure décrite dans cette tâche explique comment tester un système de fichiers pour vous aider à déterminer si un système de fichiers non répertorié conserve l'intégrité des données.

La reprise en ligne d'un gestionnaire de files d'attente multi-instance peut être déclenchée par des pannes matérielles ou logicielles, y compris des problèmes de mise en réseau qui empêchent le gestionnaire de files d'attente d'écrire dans ses données ou fichiers journaux. Vous souhaitez principalement provoquer des incidents sur le serveur de fichiers. Mais vous devez également provoquer l'échec des serveurs IBM MQ, afin de tester la libération des verrous. Pour avoir confiance dans un système de fichiers partagé, testez tous les incidents suivants et tous les autres incidents spécifiques à votre environnement:

1. Arrêt du système d'exploitation sur le serveur de fichiers, y compris la synchronisation des disques.
2. Arrêt du système d'exploitation sur le serveur de fichiers sans synchronisation des disques.
3. En appuyant sur le bouton de réinitialisation sur chacun des serveurs.
4. Extraction du câble réseau de chacun des serveurs.
5. Retrait du câble d'alimentation de chacun des serveurs.
6. Mise hors tension de chacun des serveurs.

Créez le répertoire sur le stockage en réseau que vous allez utiliser pour partager les données et les journaux du gestionnaire de files d'attente. Le propriétaire du répertoire doit être un administrateur IBM MQ ou, en d'autres termes, un membre du groupe mqm sur AIX and Linux. L'utilisateur qui exécute les tests doit disposer des droits d'administrateur IBM MQ.

Utilisez l'exemple d'exportation et de montage d'un système de fichiers dans [Création d'un gestionnaire de files d'attente multi-instance sur Linux](#) ou [Création d'un gestionnaire de files d'attente multi-instance à l'aide de la mise en miroir du journal et NetServer sur IBM i](#) pour vous aider à configurer le système de fichiers. Les différents systèmes de fichiers requièrent des étapes de configuration différentes. Lisez la documentation du système de fichiers.

**Remarque :** Exécutez l'exemple de programme IBM MQ MQI client **amqsfhac** en parallèle avec **amqmfsc** pour démontrer qu'un gestionnaire de files d'attente maintient l'intégrité des messages en cas d'échec.

### Procédure

Dans chacune des vérifications, provoquez tous les échecs de la liste précédente pendant que le vérificateur de système de fichiers est en cours d'exécution. Si vous prévoyez d'exécuter **amqsfhac**

en même temps que **amqmfscck**, effectuez la tâche «Exécution de amqsfhac pour tester l'intégrité des messages», à la page 131 en parallèle avec cette tâche.

1. Montez le répertoire exporté sur les deux serveurs IBM MQ .

Sur le serveur de système de fichiers, créez un répertoire partagé `sharedet` un sous-répertoire pour sauvegarder les données des gestionnaires de files d'attente multi-instance, `qmdat`. Pour un exemple de configuration d'un répertoire partagé pour les gestionnaires de files d'attente multi-instance sous Linux, voir [Création d'un gestionnaire de files d'attente multi-instance sous Linux](#)

2. Vérifiez le comportement de base du système de fichiers.

Sur un serveur IBM MQ , exécutez le vérificateur de système de fichiers sans paramètres.

Sur le serveur IBM MQ 1:

```
amqmfscck /shared/qmdat
```

3. Vérifiez l'écriture simultanée dans le même répertoire à partir des deux serveurs IBM MQ .

Sur les deux serveurs IBM MQ , exécutez le vérificateur de système de fichiers en même temps avec l'option `-c` .

Sur le serveur IBM MQ 1:

```
amqmfscck -c /shared/qmdat
```

Sur le serveur IBM MQ 2:

```
amqmfscck -c /shared/qmdat
```

4. Vérifiez l'attente et la libération des verrous sur les deux serveurs IBM MQ .

Sur les deux serveurs IBM MQ , exécutez le vérificateur de système de fichiers en même temps avec l'option `-w` .

Sur le serveur IBM MQ 1:

```
amqmfscck -w /shared/qmdat
```

Sur le serveur IBM MQ 2:

```
amqmfscck -w /shared/qmdat
```

5. Vérifiez l'intégrité des données.

- a) Formatez le fichier de test.

Créez un fichier volumineux dans le répertoire testé. Le fichier est formaté pour que les phases suivantes puissent aboutir. Le fichier doit être suffisamment volumineux pour qu'il y ait suffisamment de temps pour interrompre la deuxième phase afin de simuler la reprise en ligne. Essayez la valeur par défaut de 262144 pages (1 Go). Le programme réduit automatiquement cette valeur par défaut sur les systèmes de fichiers lents de sorte que le formatage se termine en environ 60 secondes

Sur le serveur IBM MQ 1:

```
amqmfscck -f /shared/qmdat
```

Le serveur répond avec les messages suivants:

```
Formatting test file for data integrity test.
```

Test file formatted with 262144 pages of data.

- b) Ecrire des données dans le fichier de test à l'aide du vérificateur de système de fichiers tout en provoquant un échec.

Exécutez le programme de test sur deux serveurs en même temps. Démarrez le programme de test sur le serveur qui va subir l'échec, puis démarrez le programme de test sur le serveur qui va survivre à l'échec. Cause de l'échec que vous recherchez.

Le premier programme de test s'arrête avec un message d'erreur. Le deuxième programme de test obtient le verrou sur le fichier de test et écrit les données dans le fichier de test en commençant là où le premier programme de test s'est arrêté. Laissez le deuxième programme de test s'exécuter jusqu'à la fin.

<i>Tableau 10. Exécution du contrôle d'intégrité des données sur deux serveurs en même temps</i>	
<b>IBM MQ Serveur 1</b>	<b>IBM MQ Serveur 2</b>
amqmfscck -a /shared/qmdata	
<p>Please start this program on a second machine with the same parameters.</p> <p>File lock acquired.</p> <p>Start a second copy of this program with the same parameters on another server.</p> <p>Writing data into test file.</p> <p>To increase the effectiveness of the test, interrupt the writing by ending the process, temporarily breaking the network connection to the networked storage, rebooting the server or turning off the power.</p>	<p>amqmfscck -a /shared/qmdata</p> <p>Waiting for lock...</p> <p>Waiting for lock...</p> <p>Waiting for lock...</p> <p>Waiting for lock...</p> <p>Waiting for lock...</p> <p>Waiting for lock...</p>
Turn the power off here.	
	<p>File lock acquired.</p> <p>Reading test file</p> <p>Checking the integrity of the data read.</p> <p>Appending data into the test file after data already found.</p> <p>The test file is full of data. It is ready to be inspected for data integrity.</p>



La durée du test dépend du comportement du système de fichiers. Par exemple, il faut généralement de 30 à 90 secondes à un système de fichiers pour libérer les verrous de fichier obtenus par le premier programme suite à une coupure de courant. Si vous avez trop peu de temps pour introduire l'échec avant que le premier programme de test ait rempli le fichier, utilisez l'option -x de **amqmfscck** pour supprimer le fichier de test. Essayez le test à partir du début avec un fichier de test plus grand.

c) Vérifiez l'intégrité des données dans le fichier de test.

Sur le serveur IBM MQ 2:

```
amqmfscck -i /shared/qmdata
```

Le serveur répond avec les messages suivants:

```
File lock acquired

Reading test file checking the integrity of the data read.

The data read was consistent.

The tests on the directory completed successfully.
```

6. Supprimez les fichiers de test.

Sur le serveur IBM MQ 2:

```
amqmfscck -x /shared/qmdata
Test files deleted.
```

Le serveur répond avec le message suivant:

```
Test files deleted.
```

## Résultats

Le programme renvoie un code d'exit égal à zéro si les tests aboutissent et différent de zéro dans le cas contraire.

## Exemples

Le premier ensemble de trois exemples montre la commande produisant une sortie minimale.

### Test réussi du verrouillage de fichier de base sur un serveur

```
> amqmfscck /shared/qmdata
The tests on the directory completed successfully.
```

### Echec du test du verrouillage de fichier de base sur un serveur

```
> amqmfscck /shared/qmdata
AMQ6245: Error Calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck' error '2'.
```

## Test réussi du verrouillage sur deux serveurs

Tableau 11. Verrouillage réussi sur deux serveurs	
IBM MQ Serveur 1	IBM MQ Serveur 2
<pre>&gt; amqmfscck -w /shared/qmdata Please start this program on a second machine with the same parameters. Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>&gt; amqmfscck -w /shared/qmdata Waiting for lock...</pre>
<pre>[ Return pressed ] Lock released.</pre>	
	<pre>Lock acquired. The tests on the directory completed successfully</pre>

Le deuxième ensemble de trois exemples montre les mêmes commandes en mode prolix.

### Test réussi du verrouillage de fichier de base sur un serveur

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")'
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd1 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd1, F_SETLK, F_RDLCK)
System call: fd2 = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd2, F_SETLK, F_RDLCK)
System call: close(fd2)
System call: write(fd1)
System call: close(fd1)
The tests on the directory completed successfully.
```

### Echec du test du verrouillage de fichier de base sur un serveur

```
> amqmfscck -v /shared/qmdata
System call: stat("/shared/qmdata")
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fchmod(fd, 0666)
System call: fstat(fd)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: write(fd)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_WRLCK)
System call: close(fd)
System call: fd = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fd, F_SETLK, F_RDLCK)
System call: fdSameFile = open("/shared/qmdata/amqmfscck.lck", O_RDWR, 0666)
System call: fcntl(fdSameFile, F_SETLK, F_RDLCK)
System call: close(fdSameFile)
System call: write(fd)
AMQxxxx: Error calling 'write()[2]' on file '/shared/qmdata/amqmfscck.lck', errno 2
(Permission denied).
```

## Test réussi du verrouillage sur deux serveurs

Tableau 12. Verrouillage réussi sur deux serveurs-mode proluxe	
IBM MQ Serveur 1	IBM MQ Serveur 2
<pre>&gt; amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL   O_CREAT   O_RDWR, 0666)' Calling 'fchmod(fd, 0666)' Calling 'fstat(fd)' Please start this program on a second machine with the same parameters. Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. Press Return or terminate the program to release the lock.</pre>	
	<pre>&gt; amqmfscck -wv /shared/qmdata Calling 'stat("/shared/qmdata")' Calling 'fd = open("/shared/qmdata/ amqmfscck.lkw", O_EXCL   O_CREAT   O_RDWR,0666)' Calling 'fd = open("/shared/qmdata/amqmfscck.lkw, O_RDWR, 0666)' Calling 'fcntl(fd, F_SETLK, F_WRLCK)' 'Waiting for lock...</pre>
<pre>[ Return pressed ] Calling 'close(fd)' Lock released.</pre>	
	<pre>Calling 'fcntl(fd, F_SETLK, F_WRLCK)' Lock acquired. The tests on the directory completed successfully</pre>

### Référence associée

Exemples de programmes à haute disponibilité

#### **Multi** Exécution de **amqsfhac** pour tester l'intégrité des messages

Exécutez l' IBM MQ MQI client exemple de programme **amqsfhac** en parallèle avec **amqmfscck** pour démontrer qu'un gestionnaire de files d'attente maintient l'intégrité des messages en cas d'échec.

### Avant de commencer

Vous avez besoin de quatre serveurs pour ce test. Deux serveurs pour le gestionnaire de files d'attente multi-instance, un pour le système de fichiers et un pour l'exécution de **amqsfhac** en tant qu'application IBM MQ MQI client .

Suivez l'étape «1», à la page 127 dans «Vérification du comportement du système de fichiers partagé sur Multiplatforms», à la page 126 pour configurer le système de fichiers pour un gestionnaire de files d'attente multi-instance.

### Pourquoi et quand exécuter cette tâche

L' IBM MQ MQI client exemple de programme **amqsfhac** vérifie qu'un gestionnaire de files d'attente utilisant le stockage en réseau conserve l'intégrité des données suite à une défaillance. Exécutez **amqsfhac** en parallèle avec **amqmfscck** pour démontrer qu'un gestionnaire de files d'attente conserve l'intégrité des messages en cas d'échec.

## Procédure

1. Créez un gestionnaire de files d'attente multi-instance sur un autre serveur, QM1, à l'aide du système de fichiers que vous avez créé à l'étape «1», à la page 127 dans [Procédure](#).  
Voir [Création d'un gestionnaire de files d'attente multi-instance](#).
2. Démarrez le gestionnaire de files d'attente sur les deux serveurs en le rendant hautement disponible.

Sur le serveur 1:

```
strmqm -x QM1
```

Sur le serveur 2:

```
strmqm -x QM1
```

3. Configurez la connexion client pour exécuter **amqsfhac**.
  - a) Utilisez la procédure décrite dans *Vérification d'une installation IBM MQ* pour la ou les plateformes utilisées par votre entreprise pour configurer une connexion client, ou les exemples de script dans [Exemples de client reconnectable](#).
  - b) Modifiez le canal client pour qu'il ait deux adresses IP, correspondant aux deux serveurs exécutant QM1.

Dans l'exemple de script, modifiez:

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('LOCALHOST(2345)') QMNAME(QM1) REPLACE
```

A :

```
DEFINE CHANNEL(CHANNEL1) CHLTYPE(CLNTCONN) TRPTYPE(TCP) +  
CONNAME('server1(2345),server2(2345)') QMNAME(QM1) REPLACE
```

où `server1` et `server2` sont les noms d'hôte des deux serveurs et 2345 est le port sur lequel le programme d'écoute de canal est en mode écoute. La valeur par défaut est généralement 1414. Vous pouvez utiliser 1414 avec la configuration du programme d'écoute par défaut.

4. Créez deux files d'attente locales sur QM1 pour le test.  
Exécutez le script MQSC suivant:

```
DEFINE QLOCAL(TARGETQ) REPLACE  
DEFINE QLOCAL(SIDEQ) REPLACE
```

5. Test de la configuration avec **amqsfhac**

```
amqsfhac QM1 TARGETQ SIDEQ 2 2 2
```

6. Testez l'intégrité des messages pendant que vous testez l'intégrité du système de fichiers.

Exécutez **amqsfhac** lors de l'étape «5», à la page 127 de [«Vérification du comportement du système de fichiers partagé sur Multiplatforms»](#), à la page 126.

```
amqsfhac QM1 TARGETQ SIDEQ 10 20 0
```

Si vous arrêtez l'instance de gestionnaire de files d'attente active, **amqsfhac** se reconnecte à l'autre instance de gestionnaire de files d'attente une fois qu'elle est devenue active. Redémarrez à nouveau l'instance de gestionnaire de files d'attente arrêtée afin de pouvoir inverser l'échec lors du prochain test. Vous devrez probablement augmenter le nombre d'itérations en fonction de l'expérimentation

avec votre environnement afin que le programme de test s'exécute suffisamment longtemps pour que la reprise en ligne se produise.

## Résultats

Un exemple d'exécution de **amqsfhac** à l'étape «6», à la page 132 est illustré dans l'exemple suivant. Dans cet exemple, le test est un succès.

```
Sample AMQSFHAC start
qmname = QM1
qname = TARGETQ
sidename = SIDEQ
transize = 10
iterations = 20
verbose = 0
Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Resolving MQRC_CALL_INTERRUPTED
MQGET browse side tranid=14 pSideinfo->tranid=14
Resolving to committed
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11
Iteration 12
Iteration 13
Iteration 14
Iteration 15
Iteration 16
Iteration 17
Iteration 18
Iteration 19
Sample AMQSFHAC end
```

Si le test a détecté un problème, la sortie signale l'échec. Dans certaines exécutions de test, MQRC\_CALL\_INTERRUPTED peut signaler "Resolving to backed out". Cela ne change rien au résultat. Le résultat varie selon que l'écriture sur le disque a été validée par le stockage de fichiers en réseau avant ou après l'échec.

### Référence associée

**amqmfsc** (vérification du système de fichiers)

[Exemples de programmes à haute disponibilité](#)

Multi

## Partage de fichiers IBM MQ sur Multiplatforms

Certains fichiers IBM MQ sont accessibles exclusivement par un gestionnaire de files d'attente actif, d'autres sont partagés.

Les fichiers IBM MQ sont divisés en fichiers programme et fichiers de données. Les fichiers programme sont généralement installés en local sur chaque serveur exécutant IBM MQ. Les gestionnaires de files d'attente partagent l'accès aux fichiers de données et aux répertoires du répertoire de données par défaut. Ils ont besoin d'un accès exclusif à leurs propres arborescences de répertoires de gestionnaires de files d'attente contenues dans chacun des répertoires qmgrs et log indiqués dans le [Figure 32](#), à la page 134.

[Figure 32](#), à la page 134 est une vue de haut niveau de la structure de répertoires IBM MQ. Il affiche les répertoires qui peuvent être partagés entre les gestionnaires de files d'attente et rendus distants. Les détails varient en fonction de la plateforme. Les lignes en pointillés indiquent les chemins configurables.

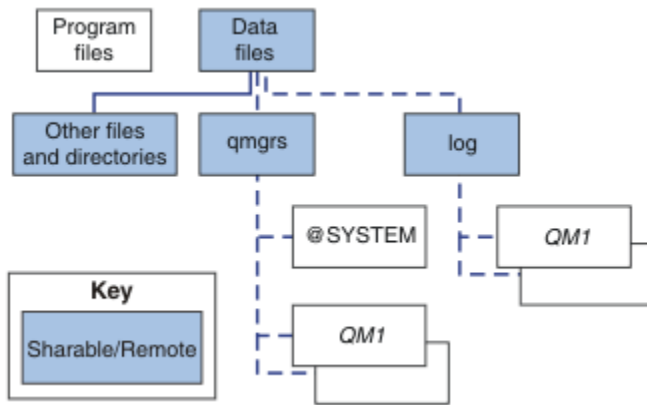


Figure 32. Vue d'ensemble de la structure de répertoire IBM MQ

### Fichiers programme

Le répertoire des fichiers programme est généralement laissé à l'emplacement par défaut, il est local et partagé par tous les gestionnaires de files d'attente sur le serveur.

### Fichiers de données

Le répertoire des fichiers de données est généralement local dans l'emplacement par défaut, /var/mqm sur les systèmes AIX and Linux et configurable sur l'installation sous Windows. Il est partagé entre les gestionnaires de files d'attente. Vous pouvez rendre l'emplacement par défaut distant, mais ne le partagez pas entre les différentes installations d' IBM MQ. L'attribut DefaultPrefix de la configuration IBM MQ pointe vers ce chemin.

### qmgrs

Il existe deux façons de spécifier l'emplacement des données du gestionnaire de files d'attente.

#### Utilisation de l'attribut Prefix

L'attribut **Prefix** indique l'emplacement du répertoire qmgrs . IBM MQ construit le nom du répertoire du gestionnaire de files d'attente à partir du nom du gestionnaire de files d'attente et le crée en tant que sous-répertoire du répertoire qmgrs .

L'attribut **Prefix** se trouve dans la strophe `QueueManager` du fichier `mqm.ini` et est hérité de la valeur de l'attribut **DefaultPrefix** de la strophe `All Queue Managers`. Par défaut, pour des raisons de simplicité administrative, les gestionnaires de files d'attente partagent généralement le même répertoire qmgrs .

Si vous modifiez l'emplacement du répertoire qmgrs pour un gestionnaire de files d'attente, vous devez modifier la valeur de son attribut **Prefix** .

L'attribut **Prefix** du répertoire QM1 dans [Figure 32](#), à la page 134 pour une plateforme AIX and Linux est le suivant:

```
Prefix=/var/mqm
```

#### Utilisation de l'attribut DataPath

L'attribut **DataPath** indique l'emplacement du répertoire de données du gestionnaire de files d'attente.

L'attribut **DataPath** indique le chemin complet, y compris le nom du répertoire de données du gestionnaire de files d'attente. L'attribut **DataPath** est différent de l'attribut **Prefix** , qui spécifie un chemin d'accès incomplet au répertoire de données du gestionnaire de files d'attente.

L'attribut **DataPath** , s'il est spécifié, se trouve dans la section `QueueManager` du fichier `mqm.ini` . S'il a été spécifié, il est prioritaire sur toute valeur de l'attribut **Prefix** .

Si vous modifiez l'emplacement du répertoire de données du gestionnaire de files d'attente pour un gestionnaire de files d'attente, vous devez modifier la valeur de l'attribut `DataPath`.

L'attribut `DataPath` du répertoire QM1 dans [Figure 32](#), à la page 134, pour une plateforme Linux ou AIX, est le suivant:

```
DataPath=/var/mqm/qmgrs/QM1
```

## Log

Le répertoire de journaux est spécifié séparément pour chaque gestionnaire de files d'attente dans la [strophe de journal](#) de la configuration du gestionnaire de files d'attente. La configuration du gestionnaire de files d'attente se trouve dans `qm.ini`.

## Sous-répertoires `DataPath/QmgrName/@IPCC`

Les sous-répertoires `DataPath/QmgrName/@IPCC` se trouvent dans le chemin de répertoire partagé. Ils sont utilisés pour construire le chemin de répertoire pour les objets du système de fichiers IPC. Ils doivent distinguer l'espace de nom d'un gestionnaire de files d'attente lorsqu'un gestionnaire de files d'attente est partagé entre des systèmes.

Les objets du système de fichiers IPC doivent être distingués par le système. Un sous-répertoire, pour chaque système sur lequel s'exécute le gestionnaire de files d'attente, est ajouté au chemin de répertoire. Voir [Figure 33](#), à la page 135.

```
DataPath/QmgrName/@IPCC/esem/myHostName/
```

*Figure 33. Exemple de sous-répertoire IPC*

`myHostName` correspond aux 20 premiers caractères du nom d'hôte renvoyé par le système d'exploitation. Sur certains systèmes, le nom d'hôte peut comporter jusqu'à 64 caractères avant la troncature. La valeur générée de `myHostName` peut provoquer un problème pour deux raisons:

1. Les 20 premiers caractères ne sont pas uniques.
2. Le nom d'hôte est généré par un algorithme DHCP qui n'attribue pas toujours le même nom d'hôte à un système.

Dans ces cas, définissez `myHostName` à l'aide de la variable d'environnement [MQS\\_IPC\\_HOST](#); voir [Figure 34](#), à la page 135.

```
export MQS_IPC_HOST= myHostName
```

*Figure 34. Exemple: définition de `MQS_IPC_HOST`*

## Autres fichiers et répertoires

D'autres fichiers et répertoires, tels que le répertoire contenant les fichiers de trace et le journal des erreurs commun, sont normalement partagés et conservés sur le système de fichiers local.

Avec la prise en charge des systèmes de fichiers partagés, IBM MQ gère l'accès exclusif à ces fichiers à l'aide de verrous de système de fichiers. Un verrou de système de fichiers permet à une seule instance d'un gestionnaire de files d'attente particulier d'être active à la fois.

Lorsque vous démarrez la première instance d'un gestionnaire de files d'attente particulier, celui-ci devient propriétaire de son répertoire de gestionnaire de files d'attente. Si vous démarrez une deuxième instance, elle ne peut devenir propriétaire que si la première instance s'est arrêtée. Si le premier gestionnaire de files d'attente est toujours en cours d'exécution, la seconde instance ne démarre pas et signale que le gestionnaire de files d'attente est en cours d'exécution ailleurs. Si le premier gestionnaire de files d'attente s'est arrêté, le second prend le contrôle des fichiers du gestionnaire de files d'attente et devient le gestionnaire de files d'attente en cours d'exécution.

Vous pouvez automatiser la procédure du deuxième gestionnaire de files d'attente qui prend le relais du premier. Démarrez le premier gestionnaire de files d'attente avec l'option `strmqm -x` qui permet à un autre gestionnaire de files d'attente de prendre le relais. Le deuxième gestionnaire de files d'attente attend ensuite que les fichiers du gestionnaire de files d'attente soient déverrouillés avant de tenter de prendre le contrôle des fichiers du gestionnaire de files d'attente et de démarrer.

## Linux AIX Structure de répertoires sur les systèmes AIX and Linux

La structure de répertoire IBM MQ sur les systèmes AIX and Linux peut être mappée à différents systèmes de fichiers pour une gestion plus simple, de meilleures performances et une meilleure fiabilité.

Utilisez la structure de répertoire flexible d' IBM MQ pour tirer parti des systèmes de fichiers partagés afin d'exécuter des gestionnaires de files d'attente multi-instance.

Utilisez la commande `crtmqm QM1` pour créer la structure de répertoires indiquée dans Figure 35, à la page 136 , où R correspond à l'édition du produit. Il s'agit d'une structure de répertoire standard pour un gestionnaire de files d'attente créé sur un système IBM MQ . Certains répertoires, fichiers et paramètres d'attribut .ini sont omis pour plus de clarté et un autre nom de gestionnaire de files d'attente peut être modifié par la fonction de gestion. Les noms des systèmes de fichiers varient en fonction des systèmes.

Dans une installation standard, chaque gestionnaire de files d'attente que vous créez pointe vers des répertoires `log` et `qmgrs` communs sur le système de fichiers local. Dans une configuration multi-instance, les répertoires `log` et `qmgrs` se trouvent sur un système de fichiers réseau partagé avec une autre installation de IBM MQ.

La Figure 35, à la page 136 présente la configuration par défaut pour IBM MQ v7.R sur AIX où R correspond à l'édition du produit. Pour des exemples de configurations multi-instance alternatives, voir «Exemples de configuration de répertoire sur les systèmes AIX and Linux», à la page 141.

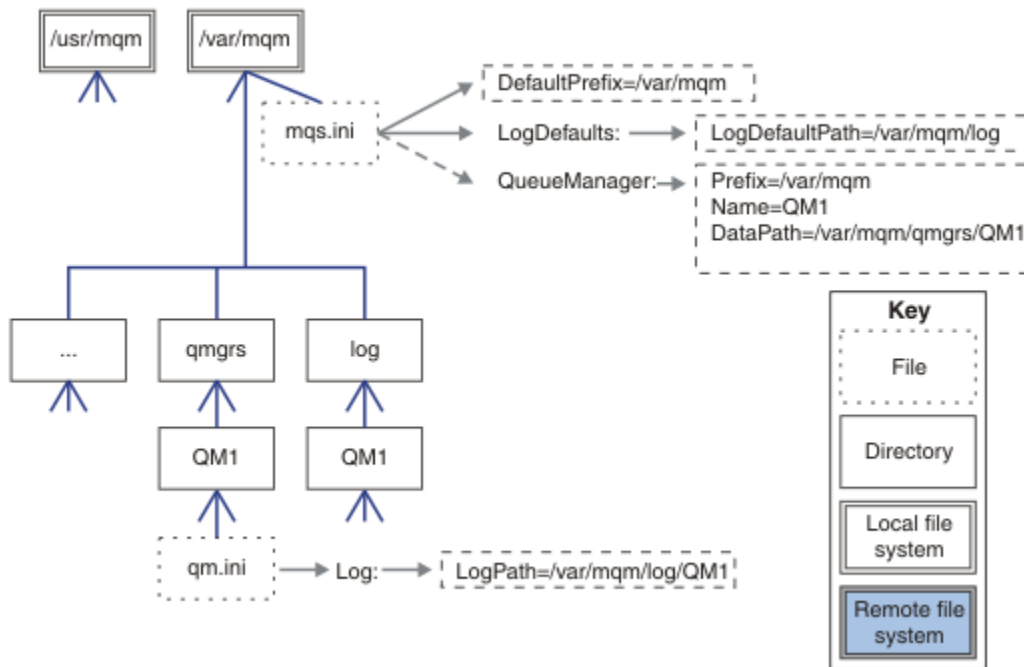


Figure 35. Exemple de structure de répertoire IBM MQ par défaut pour les systèmes AIX and Linux

Le produit est installé dans `/usr/mqm` sous AIX et `/opt/mqm` sur les autres systèmes, par défaut. Les répertoires de travail sont installés dans le répertoire `/var/mqm` .

**Remarque :** Si vous avez créé le système de fichiers `/var/mqm` avant d'installer IBM MQ , vérifiez que l'utilisateur `mqm` dispose des droits d'accès complets au répertoire, par exemple, le mode de fichier 755.



**Remarque :** Le répertoire `/var/mqm/errors` doit être un système de fichiers distinct pour empêcher les FFDC produits par le gestionnaire de files d'attente de remplir le système de fichiers qui contient `/var/mqm`.

Pour plus d'informations, voir [Création de systèmes de fichiers sur des systèmes AIX and Linux](#).

Les répertoires `log` et `qmgrs` sont affichés dans leurs emplacements par défaut définis par les valeurs par défaut des attributs `LogDefaultPath` et `DefaultPrefix` dans le fichier `mqs.ini`. Lorsqu'un gestionnaire de files d'attente est créé, le répertoire de données du gestionnaire de files d'attente est créé par défaut dans `DefaultPrefix/qmgrset` et le répertoire des fichiers journaux dans `LogDefaultPath/log`. `LogDefaultPath` et `DefaultPrefix` n'ont d'effet que lorsque des gestionnaires de files d'attente et des fichiers journaux sont créés par défaut. L'emplacement réel d'un répertoire de gestionnaire de files d'attente est sauvegardé dans le fichier `mqs.ini` et l'emplacement du répertoire de fichiers journaux est sauvegardé dans le fichier `qm.ini`.

Le répertoire des fichiers journaux d'un gestionnaire de files d'attente est défini dans le fichier `qm.ini` de l'attribut `LogPath`. Utilisez l'option `-ld` dans la commande **`crtmqm`** pour définir l'attribut `LogPath` d'un gestionnaire de files d'attente ; par exemple, **`crtmqm -ld LogPath QM1`**. Si vous omettez le paramètre `ld`, la valeur de `LogDefaultPath` est utilisée à la place.

Le répertoire de données du gestionnaire de files d'attente est défini dans l'attribut `DataPath` de la section `QueueManager` du fichier `mqs.ini`. Utilisez l'option `-md` dans la commande **`crtmqm`** pour définir `DataPath` pour un gestionnaire de files d'attente ; par exemple, **`crtmqm -md DataPath QM1`**. Si vous omettez le paramètre `md`, la valeur de l'attribut `DefaultPrefix` ou `Prefix` est utilisée à la place. Préfixe a priorité sur `DefaultPrefix`.

En règle générale, créez `QM1` en spécifiant les répertoires de journaux et de données dans une seule commande.

```
crtmqm
-md DataPath -ld
LogPath QM1
```

Vous pouvez modifier l'emplacement d'un journal de gestionnaire de files d'attente et les répertoires de données d'un gestionnaire de files d'attente existant en éditant les attributs `DataPath` et `LogPath` dans le fichier `qm.ini` lorsque le gestionnaire de files d'attente est arrêté.

Le chemin d'accès au répertoire `errors`, comme les chemins d'accès à tous les autres répertoires dans `/var/mqm`, n'est pas modifiable. Toutefois, les répertoires peuvent être montés sur des systèmes de fichiers différents ou liés symboliquement à des répertoires différents.

Linux

AIX

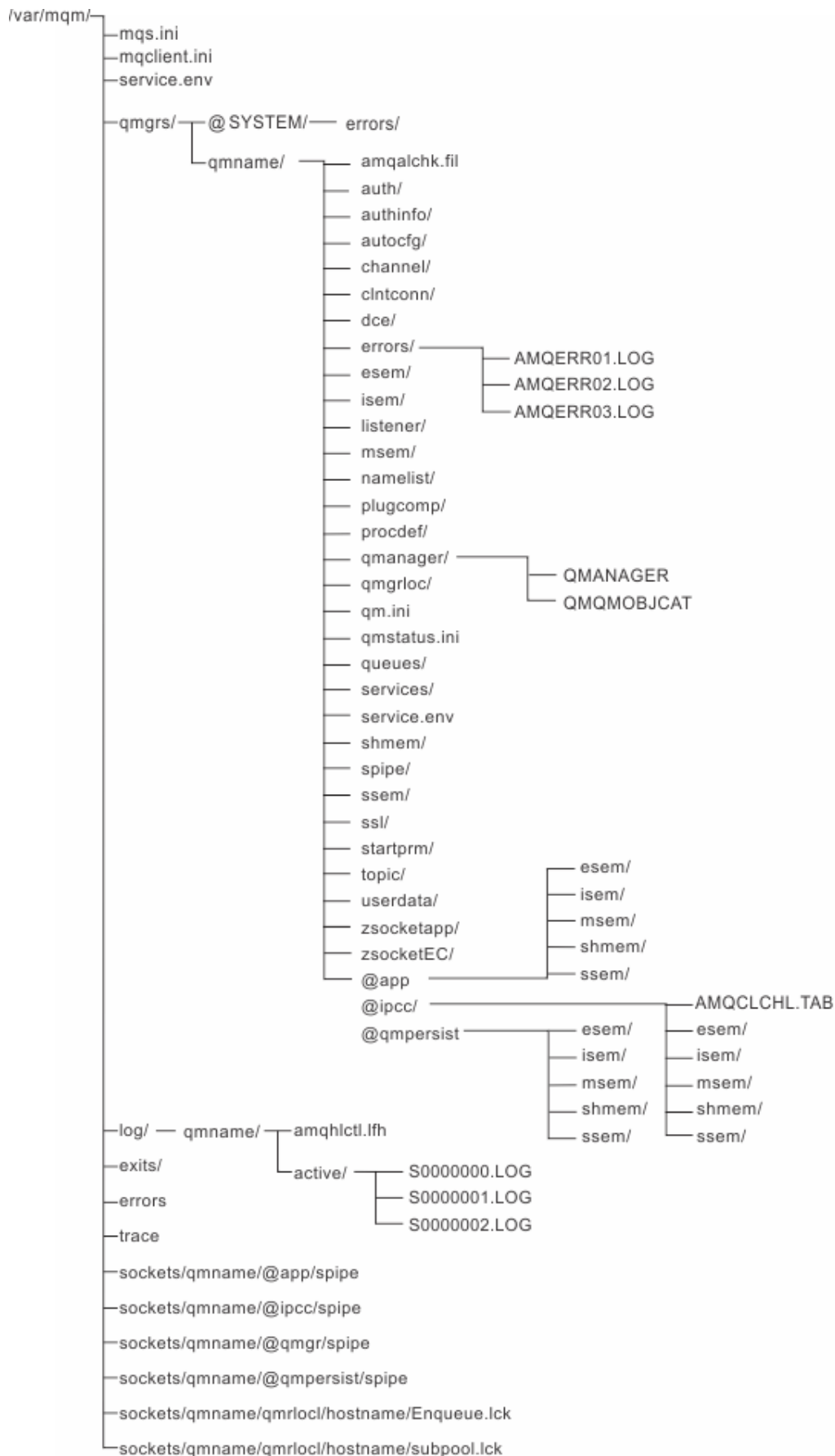
## **Contenu du répertoire sur les systèmes AIX and Linux**

Contenu des répertoires associés à un gestionnaire de files d'attente.

Pour plus d'informations sur l'emplacement des fichiers du produit, voir [Choix d'un emplacement d'installation](#)

Pour plus d'informations sur les autres configurations de répertoire, voir [«Planification de la prise en charge du système de fichiers sur Multiplatforms»](#), à la page 123.

La structure de répertoire suivante est représentative de IBM MQ lorsqu'un gestionnaire de files d'attente est utilisé depuis un certain temps. La structure réelle dont vous disposez dépend des opérations qui ont eu lieu sur le gestionnaire de files d'attente.



## **/var/mqm/**

Le répertoire `/var/mqm` contient les fichiers de configuration et les répertoires de sortie qui s'appliquent à une installation IBM MQ dans son ensemble, et non à un gestionnaire de files d'attente individuel.

Tableau 13. Contenu documenté du répertoire `/var/mqm` sous AIX and Linux

<b>Nom de répertoire ou de fichier</b>	<b>Contenu</b>
<b><u>mqs.ini</u></b>	Fichier de configuration à l'échelle de l'installation IBM MQ , lu lorsqu'un gestionnaire de files d'attente démarre. Chemin de fichier modifiable à l'aide de la variable d'environnement <b>AMQ_MQS_INI_LOCATION</b> . Vérifiez qu'il est défini et exporté dans le shell dans lequel la commande <b>strmqm</b> est exécutée.
<b><u>mqclient.ini</u></b>	Fichier de configuration client par défaut lu par les programmes IBM MQ MQI client . Chemin de fichier modifiable à l'aide de la variable d'environnement <b>MQCLNTCF</b> .
<b><u>service.env</u></b>	Contient des variables d'environnement de portée machine pour un processus de service. Chemin d'accès au fichier corrigé.
<b><u>erreurs /</u></b>	Journaux d'erreurs de portée machine et fichiers FFST . Chemin de répertoire corrigé. Voir aussi <a href="#">FFST: systèmes IBM MQ for UNIX et Linux</a> .
<b><u>sockets /</u></b>	Contient des informations pour chaque gestionnaire de files d'attente à utiliser uniquement par le système.
<b><u>trace /</u></b>	Fichiers de trace. Chemin de répertoire corrigé.
<b><u>Web /</u></b>	Répertoire du serveur mqweb.
<b><u>exits /</u></b>	Répertoire par défaut contenant les programmes d'exit de canal utilisateur. Emplacement modifiable dans les strophes ApiExit du fichier <code>mqs.ini</code> .
<b><u>exits64/</u></b>	

## **/var/mqm/qmgrs/qmname/**

`/var/mqm/qmgrs/qmname/` contient les répertoires et les fichiers d'un gestionnaire de files d'attente. Le répertoire est verrouillé pour un accès exclusif par l'instance de gestionnaire de files d'attente active. Le chemin de répertoire est directement modifiable dans le fichier `mqs.ini` ou à l'aide de l'option **md** de la commande **crtmqm** .

Tableau 14. Contenu documenté du répertoire `/var/mqm/qmgrs/qmname` sous AIX and Linux

<b>Nom de répertoire ou de fichier</b>	<b>Contenu</b>
<b><u>qm.ini</u></b>	Fichier de configuration du gestionnaire de files d'attente, lu lorsqu'un gestionnaire de files d'attente démarre.

Tableau 14. Contenu documenté du répertoire /var/mqm/qmgrs/qmname sous AIX and Linux (suite)

Nom de répertoire ou de fichier	Contenu
<u>erreurs /</u>	Journaux des erreurs de portée du gestionnaire de files d'attente. qmname = @system contient des messages relatifs aux canaux pour un gestionnaire de files d'attente inconnu ou non disponible.
<u>@ipcc/AMQCLCHL.TAB</u>	Table de contrôle des canaux client par défaut, créée par le serveur IBM MQ et lue par les programmes IBM MQ MQI client . Chemin de fichier modifiable à l'aide des variables d'environnement <b>MQCHLLIB</b> et <b>MQCHLTAB</b> .
<b>QMANAGER</b>	Fichier objet du gestionnaire de files d'attente: QMANAGER Catalogue d'objets du gestionnaire de files d'attente: QMQMOBJCAT
<u>authinfo /</u>	Chaque objet défini dans le gestionnaire de files d'attente est associé à un fichier dans ces répertoires. Le nom de fichier correspond approximativement au nom de la définition ; voir <u>Description des noms de fichier IBM MQ</u> .
<u>canal /</u>	
<u>clntconn /</u>	
<u>programme d'écoute /</u>	
<u>liste de noms /</u>	
<u>procdef /</u>	
<u>files d'attente /</u>	
<u>services /</u>	
<u>rubriques /</u>	
...	Autres répertoires utilisés par IBM MQ, tels que @ipcc, à modifier uniquement par IBM MQ.
<b>données utilisateur /</b>	Peut être utilisé pour stocker l'état persistant des applications (peut être utilisé par le gestionnaire de files d'attente de données répliquées lors du déplacement des gestionnaires de files d'attente vers des noeuds différents-voir <u>Stockage de l'état des applications persistantes</u> .)
<b>DataPath\autocfg</b>	Utilisé pour la configuration automatique

### **/var/mqm/log/qmname /**

/var/mqm/log/qmname/ contient les fichiers journaux du gestionnaire de files d'attente. Le répertoire est verrouillé pour un accès exclusif par l'instance de gestionnaire de files d'attente active. Le chemin de répertoire est modifiable dans le fichier qm.ini ou à l'aide de l'option **ld** de la commande **crtmqm** .

Tableau 15. Contenu documenté du répertoire /var/mqm/log/qmname sous AIX and Linux

Nom de répertoire ou de fichier	Contenu
<b>amqhlctl.lfh</b>	Fichier de contrôle du journal.
<b>active /</b>	Ce répertoire contient les fichiers journaux portant le numéro S0000000.LOG, S0000001.LOG, S0000002.LOG, etc.

## /opt/mqm

/opt/mqm est, par défaut, le répertoire d'installation sur la plupart des plateformes. Voir «Espace disque requis sur Multiplatforms», à la page 120 pour plus d'informations sur la quantité d'espace dont vous avez besoin pour le répertoire d'installation sur la ou les plateformes utilisées par votre entreprise.

Linux

AIX

## Exemples de configuration de répertoire sur les systèmes AIX and Linux

Exemples de configurations de système de fichiers alternatives sur les systèmes AIX and Linux .

Vous pouvez personnaliser la structure de répertoires IBM MQ de différentes manières pour atteindre un certain nombre d'objectifs différents.

- Placez les répertoires `qmgrs` et `log` sur des systèmes de fichiers partagés distants pour configurer un gestionnaire de files d'attente multi-instance.
- Utilisez des systèmes de fichiers distincts pour les répertoires de données et de journaux et allouez les répertoires à différents disques afin d'améliorer les performances en réduisant les conflits d'E-S.
- Utilisez des unités de stockage plus rapides pour les répertoires qui ont un plus grand impact sur les performances. Le temps d'attente des périphériques physiques est souvent un facteur plus important dans les performances de la messagerie persistante que lorsqu'un périphérique est monté en local ou à distance. La liste suivante répertorie les répertoires les plus et les moins sensibles aux performances.

1. `log`

2. `qmgrs`

3. Autres répertoires, y compris `/usr/mqm`

- Créez les répertoires `qmgrs` et `log` sur les systèmes de fichiers qui sont alloués au stockage avec une bonne résilience, par exemple une grappe de disques redondante.
- Il est préférable de stocker les journaux d'erreurs communs dans `var/mqm/errors`, en local, plutôt que sur un système de fichiers réseau, afin que les erreurs liées au système de fichiers réseau puissent être consignées.

Figure 36, à la page 142 est un modèle à partir duquel des structures de répertoire IBM MQ alternatives sont dérivées. Dans le modèle, les lignes en pointillés représentent les chemins configurables. Dans les exemples, les lignes en pointillés sont remplacées par des lignes continues qui correspondent aux informations de configuration stockées dans la variable d'environnement `AMQ_MQS_INI_LOCATION` et dans les fichiers `mqs.ini` et `qm.ini`.

**Remarque :** Les informations de chemin sont affichées telles qu'elles apparaissent dans les fichiers `mqs.ini` ou `qm.ini`. Si vous fournissez des paramètres de chemin dans la commande `crtmqm`, omettez le nom du répertoire du gestionnaire de files d'attente: le nom du gestionnaire de files d'attente est ajouté au chemin par IBM MQ.

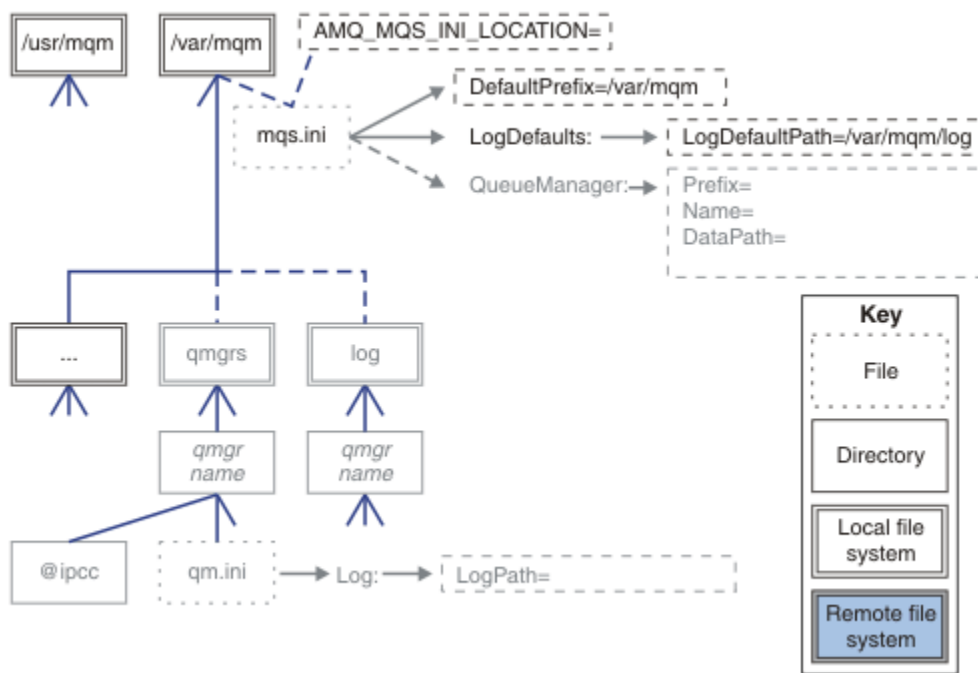


Figure 36. Modèle de canevas de structure de répertoire

### Structure de répertoire standard pour IBM MQ

Figure 37, à la page 143 est la structure de répertoire par défaut créée dans IBM MQ à l'aide de la commande **crtmqm QM1**.

Le fichier **mqs.ini** comporte une section pour le gestionnaire de files d'attente QM1, créée en faisant référence à la valeur de **DefaultPrefix**. La section **Log** du fichier **qm.ini** comporte une valeur pour **LogPath**, définie par référence à **LogDefaultPath** dans **mqs.ini**.

Utilisez les paramètres **crtmqm** facultatifs pour remplacer les valeurs par défaut de **DataPath** et **LogPath**.

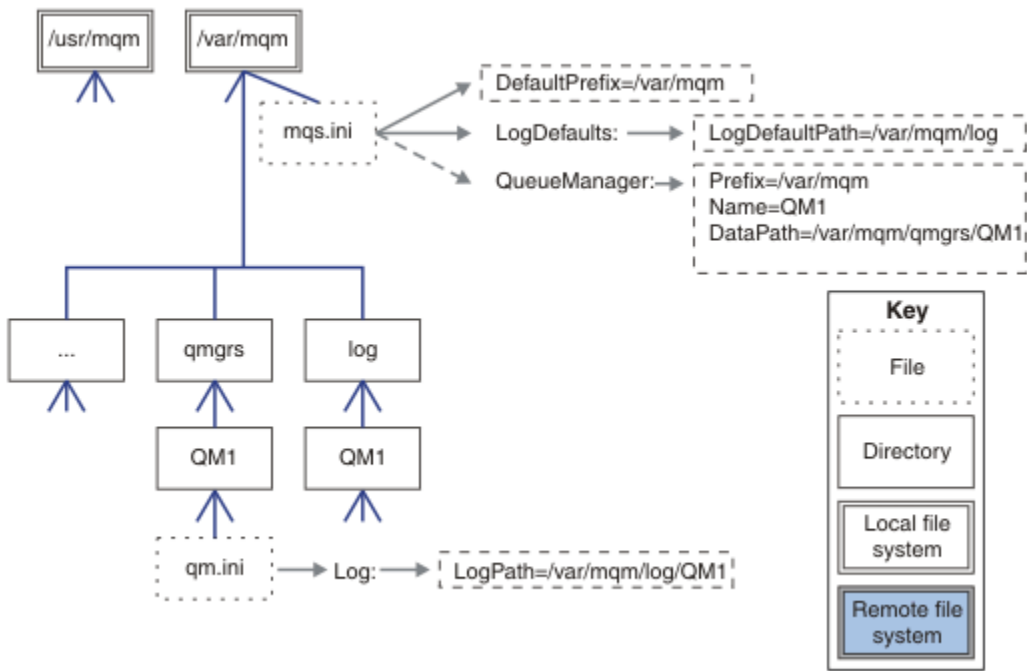


Figure 37. Exemple de structure de répertoire IBM MQ par défaut pour les systèmes AIX and Linux

### Partage des répertoires qmgrs et log par défaut

Une alternative à «Tout partager», à la page 144 consiste à partager les répertoires `qmgrs` et `log` séparément ( Figure 38, à la page 143 ). Dans cette configuration, il n'est pas nécessaire de définir `AMQ_MQS_INI_LOCATION` car la valeur par défaut `mqs.ini` est stockée dans le système de fichiers `/var/mqm` local. Les fichiers et répertoires, tels que `mqlclient.ini` et `mqserver.ini`, ne sont pas non plus partagés.

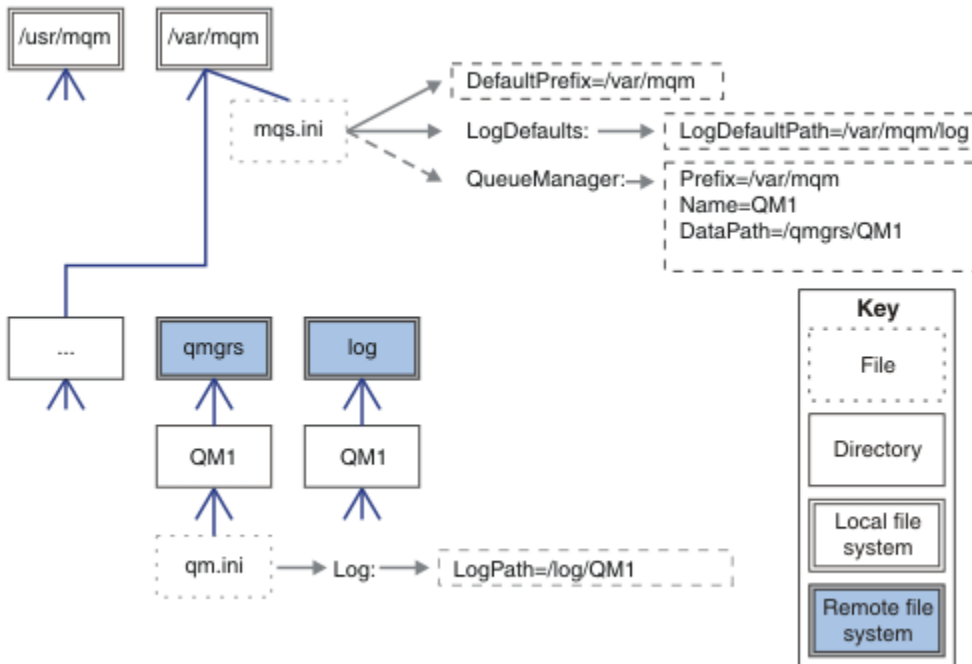


Figure 38. Partage des répertoires qmgrs et Log

## Partage des répertoires qmgrs et log

La configuration dans Figure 39, à la page 144 place log et qmgrs dans un système de fichiers partagé distant commun nommé /ha. La même configuration physique peut être créée de deux manières différentes.

1. Définissez `LogDefaultPath=/ha`, puis exécutez la commande `crtmqm - md /ha/qmgrs QM1`. Le résultat est exactement comme illustré dans la Figure 39, à la page 144.
2. Laissez les chemins par défaut inchangés, puis exécutez la commande `crtmqm - ld /ha/log - md /ha/qmgrs QM1`.

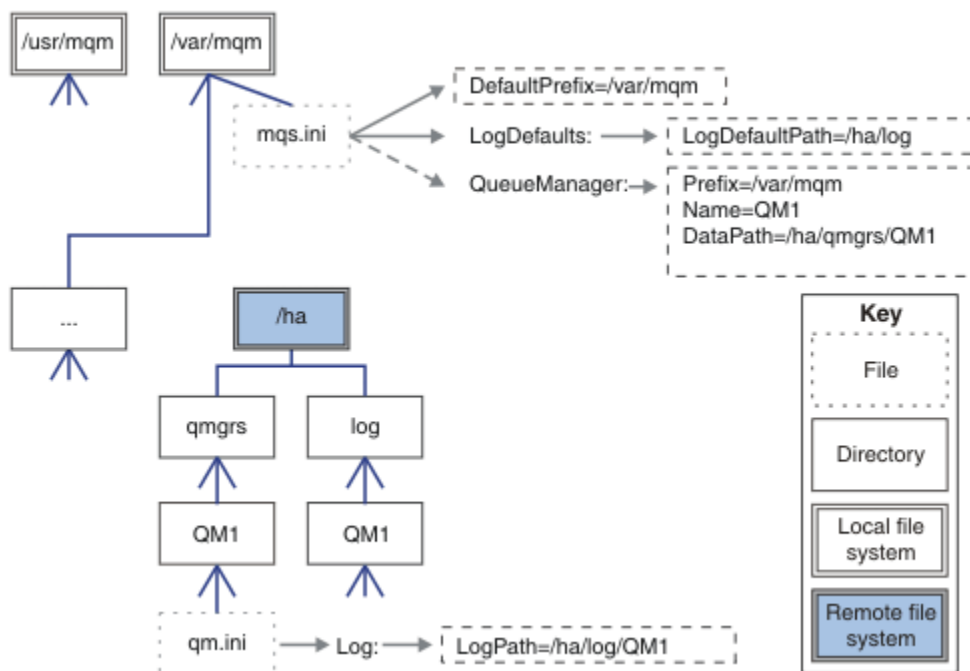


Figure 39. Partage des répertoires qmgrs et log

## Tout partager

Figure 40, à la page 145 est une configuration simple pour un système avec un stockage de fichiers en réseau.

Montez `/var/mqm` en tant que système de fichiers partagé distant. Par défaut, lorsque vous démarrez `QM1`, il recherche `/var/mqm`, le trouve sur le système de fichiers partagé et lit le fichier `mqs.ini` dans `/var/mqm`. Au lieu d'utiliser le fichier `/var/mqm/mqs.ini` unique pour les gestionnaires de files d'attente sur tous vos serveurs, vous pouvez définir la variable d'environnement `AMQ_MQS_INI_LOCATION` sur chaque serveur pour qu'elle pointe vers des fichiers `mqs.ini` différents.

**Remarque :** Le contenu du fichier d'erreurs générique dans `/var/mqm/errors/` est partagé entre les gestionnaires de files d'attente sur des serveurs différents.



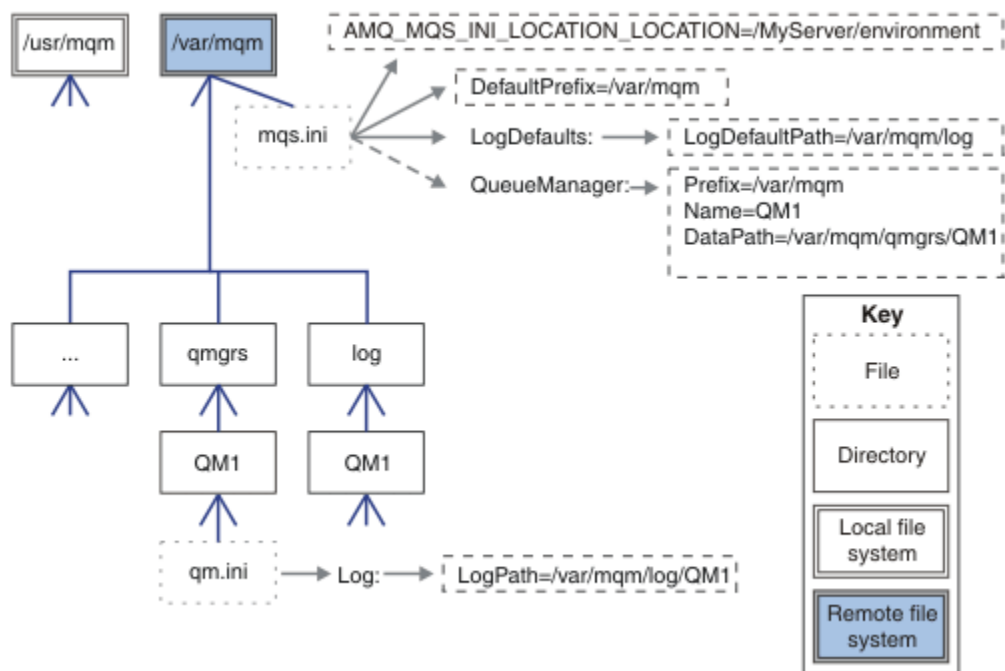


Figure 40. Tout partager

Notez que vous ne pouvez pas l'utiliser pour les gestionnaires de files d'attente multi-instance. Cela est dû au fait qu'il est nécessaire que chaque hôte d'un gestionnaire de files d'attente multi-instance dispose de sa propre copie locale de /var/mqm pour suivre les données locales, telles que les sémaphores et la mémoire partagée. Ces entités ne peuvent pas être partagées entre les hôtes.

## Windows Structure de répertoires sur les systèmes Windows

Cette rubrique explique comment rechercher des informations et des répertoires de configuration de gestionnaire de files d'attente sous Windows.

Les répertoires par défaut pour l'installation de IBM MQ for Windows sont les suivants:

### Program Directory

C:\Program Files\IBM\MQ

### Répertoire de données

C:\ProgramData\IBM\MQ

**Important :** **Windows** Pour les installations de Windows, les répertoires sont tels qu'indiqués, sauf s'il existe une installation précédente du produit qui contient toujours des entrées de registre et/ou des gestionnaires de files d'attente. Dans ce cas, la nouvelle installation utilise l'ancien emplacement du répertoire de données. Pour plus d'informations, voir [Emplacement du répertoire de programme et des données](#).

Si vous souhaitez savoir quel répertoire d'installation et quel répertoire de données est utilisé, exécutez la commande `dspmqr`.

Le répertoire d'installation est répertorié dans la zone **InstPath** et le répertoire de données est répertorié dans la zone **DataPath**.

L'exécution de la commande `dspmqr` affiche, par exemple, les informations suivantes:

```
>dspmqr
Name:      IBM MQ
Version:   9.0.0.0
Level:     p900-L160512.4
BuildType: IKAP - (Production)
Platform:  IBM MQ for Windows (x64 platform)
Mode:      64-bit
```

```

O/S: Windows 7 Professional x64 Edition, Build 7601: SP1
InstName: Installation1
InstDesc:
Primary: Yes
InstPath: C:\Program Files\IBM\MQ
DataPath: C:\ProgramData\IBM\MQ
MaxCmdLevel: 900
LicenseType: Production

```

## Gestionnaires de files d'attente multi-instance

Pour configurer un gestionnaire de files d'attente multi-instance, les répertoires de journaux et de données doivent être placés sur le stockage en réseau, de préférence sur un serveur différent de celui des serveurs qui exécutent des instances du gestionnaire de files d'attente.

Deux paramètres sont fournis dans la commande **crtmqm**, **-md** et **-ld**, pour faciliter la spécification de l'emplacement des répertoires de données et de journaux du gestionnaire de files d'attente. L'effet de la spécification du paramètre **-md** est 4 x 4:

1. La `mqs.ini` strophe `QueueManager\QmgrName` contient une nouvelle variable, `DataPath`, qui pointe vers le répertoire de données du gestionnaire de files d'attente. Contrairement à la variable `Préfixe`, le chemin inclut le nom du répertoire du gestionnaire de files d'attente.
2. Les informations de configuration du gestionnaire de files d'attente stockées dans le fichier `mqs.ini` sont réduites à `Nom`, `Préfixe`, `Répertoire` et `DataPath`.

## Windows Contenu de l'annuaire

Répertorie l'emplacement et le contenu des répertoires IBM MQ .

Une configuration IBM MQ comporte trois ensembles principaux de fichiers et de répertoires:

1. Fichiers exécutables et autres fichiers en lecture seule qui ne sont mis à jour que lorsque la maintenance est appliquée. Exemple :
  - Le fichier `Readme`
  - Le plug-in IBM MQ Explorer et les fichiers d'aide
  - Fichiers de licence

Ces fichiers sont décrits dans [Tableau 16](#), à la [page 146](#).

2. Fichiers et répertoires potentiellement modifiables qui ne sont pas spécifiques à un gestionnaire de files d'attente particulier. Ces fichiers et répertoires sont décrits dans [Tableau 17](#), à la [page 147](#).
3. Fichiers et répertoires spécifiques à chaque gestionnaire de files d'attente sur un serveur. Ces fichiers et répertoires sont décrits dans [Tableau 18](#), à la [page 148](#).

## Répertoires et fichiers de ressources

Les répertoires de ressources et les fichiers contiennent tous les codes exécutables et les ressources permettant d'exécuter un gestionnaire de files d'attente. La variable `FilePath`, dans la clé de registre de configuration IBM MQ spécifique à l'installation, contient le chemin d'accès aux répertoires de ressources.

Chemin de fichier	Contenu
<code>FilePath\bin</code>	Commandes et DLL
<code>FilePath\bin64</code>	Commandes et DLL (64 bits)
<code>FilePath\conv</code>	Tables de conversion de données
<code>FilePath\doc</code>	Fichiers d'aide de l'assistant
<code>FilePath\MQExplorer</code>	Plug-in Eclipse d'aide Explorer et Explorer
<code>FilePath\gskit8</code>	Global Security Kit

Tableau 16. Répertoires et fichiers du répertoire *FilePath* (suite)

Chemin de fichier	Contenu
<i>FilePath</i> \java	Ressources Java , y compris JRE
<i>FilePath</i> \licenses	Informations sur la licence
<i>FilePath</i> \Non_IBM_License	Informations sur la licence
<i>FilePath</i> \properties	Utilisé en interne
<i>FilePath</i> \Tivoli	
<i>FilePath</i> \tools	Ressources et exemples de développement
<i>FilePath</i> \web	Décrit dans IBM MQ Console et dans la structure des fichiers de composant d'installation REST API pour les fichiers non modifiables.
<i>FilePath</i> \Uninst	Utilisé en interne
<i>FilePath</i> \README.TXT	Fichier Readme

## Répertoires non spécifiques à un gestionnaire de files d'attente

Certains répertoires contiennent des fichiers, tels que des fichiers de trace et des journaux d'erreurs, qui ne sont pas spécifiques à un gestionnaire de files d'attente spécifique. La variable *DefaultPrefix* contient le chemin d'accès à ces répertoires. *DefaultPrefix* fait partie de la section *AllQueueManagers* .

Tableau 17. Répertoires et fichiers du répertoire *DefaultPrefix*

Chemin de fichier	Contenu
<i>DefaultPrefix</i> \config	Utilisé en interne
<i>DefaultPrefix</i> \conv	Fichier de contrôle de conversion <i>ccsid_part2.tbl</i> et <i>ccsid.tbl data</i> , décrit dans <a href="#">Conversion de données</a>
<i>DefaultPrefix</i> \errors	Journaux des erreurs autres que ceux du gestionnaire de files d'attente, <i>AMQERR nn.LOG</i>
<i>DefaultPrefix</i> \exits	Programmes d'exit de canal
<i>DefaultPrefix</i> \exits64	Programmes d'exit de canal (64 bits)
<i>DefaultPrefix</i> \ipc	Non utilisées
<i>DefaultPrefix</i> \qmgrs	Décrit dans <a href="#">Tableau 18</a> , à la page 148
<i>DefaultPrefix</i> \trace	Fichiers de trace
<i>DefaultPrefix</i> \web	Décrit dans la section Structure des fichiers de composant d'installation IBM MQ Console et REST API pour les fichiers modifiables par l'utilisateur
<i>DefaultPrefix</i> \amqmjpse.txt	Utilisé en interne

## Répertoires du gestionnaire de files d'attente

Lorsque vous créez un gestionnaire de files d'attente, un nouvel ensemble de répertoires, spécifique au gestionnaire de files d'attente, est créé.

Si vous créez un gestionnaire de files d'attente avec le paramètre **-md filepath** , le chemin est stocké dans la variable *DataPath* dans la strophe de gestionnaire de files d'attente du fichier *mqs.ini* . Si vous créez un gestionnaire de files d'attente sans définir le paramètre **-md filepath** , les répertoires du

gestionnaire de files d'attente sont créés dans le chemin stocké dans *DefaultPrefix* le chemin est copié dans la variable *Préfixe* de la section du gestionnaire de files d'attente du fichier *mqs.ini* .

<i>Tableau 18. Répertoires et fichiers dans les répertoires DataPath et Prefix\qmgrs\QmgrName</i>	
<b>Chemin de fichier</b>	<b>Contenu</b>
<i>DataPath\@ipcc</i>	Emplacement par défaut pour AMQCLCHL . TAB, la table de connexion client.
<i>DataPath\authinfo</i>	Utilisé en interne.
<i>DataPath\channel</i>	
<i>DataPath\clntconn</i>	
<i>DataPath\errors</i>	Journaux des erreurs, AMQERR <i>nn</i> . LOG
<i>DataPath\listener</i>	Utilisé en interne.
<i>DataPath\namelist</i>	
<i>DataPath\plugcomp</i>	
<i>DataPath\procdef</i>	
<i>DataPath\qmanager</i>	
<i>DataPath\queues</i>	
<i>DataPath\services</i>	
<i>DataPath\ssl</i>	
<i>DataPath\startprm</i>	
<i>DataPath\topic</i>	
<i>DataPath\active</i>	
<i>DataPath\active.dat</i>	
<i>DataPath\amqalchk.fil</i>	
<i>DataPath\master</i>	
<i>DataPath\master.dat</i>	
<i>DataPath\qm.ini</i>	Configuration du gestionnaire de files d'attente
<i>DataPath\qmstatus.ini</i>	Etat de gestionnaire de files d'attente
<i>DataPath\userdata</i>	Peut être utilisé pour stocker l'état persistant des applications.
<i>Prefix\qmgrs\QmgrName</i>	Utilisé en interne
<i>Prefix\qmgrs\@SYSTEM</i>	Non utilisées
<i>Prefix\qmgrs\@SYSTEM\errors</i>	
<i>DataPath\autocfg</i>	Utilisé pour la configuration automatique

## Structure de répertoire sous IBM i

Une description du système de fichiers intégré est fournie et la structure de répertoire du système de fichiers intégré IBM MQ est décrite pour le serveur, le client et Java.

Le système de fichiers intégré (IFS) est une partie de IBM i qui prend en charge la gestion des entrées-sorties de flux et du stockage similaire à l'ordinateur personnel, les systèmes d'exploitation AIX and Linux , tout en fournissant une structure d'intégration sur toutes les informations stockées sur le serveur.

Sous IBM i , les noms de répertoire commencent par le caractère & (ampersand) au lieu du caractère @ (at). Par exemple, @system sous IBM i est &system.

## Système de fichiers racine IFS pour le serveur IBM MQ

Lorsque vous installez IBM MQ Server for IBM i, les répertoires suivants sont créés dans le système de fichiers racine IFS.

ProdData:

### Présentation

#### QIBM

```
'-- ProdData
    '-- mqm
    '-- doc
    '-- inc
    '-- lib
    '-- samp
    '-- licenses
    '-- LicenseDoc
    '-- 5724H72_V8R0M0
```

#### **/QIBM/ProdData/mqm**

Les sous-répertoires ci-dessous contiennent toutes les données du produit, par exemple, les classes C + +, les fichiers de format de trace et les fichiers de licence. Les données de ce répertoire sont supprimées et remplacées chaque fois que le produit est installé.

#### **/QIBM/ProdData/mqm/doc**

Une référence de commande pour les commandes CL est fournie au format HTML et installée ici.

#### **/QIBM/ProdData/mqm/inc**

Fichiers d'en-tête pour la compilation de vos programmes C ou C + +.

#### **/QIBM/ProdData/mqm/lib**

Fichiers auxiliaires utilisés par MQ.

#### **/QIBM/ProdData/mqm/samp**

Autres échantillons.

#### **/QIBM/ProdData/mqm/licenses**

Fichiers de licence. Les deux fichiers de chaque langue sont nommés comme LA\_ *xx* et LI\_ *xx* , où *xx* est l'identificateur de langue à 2 caractères pour chaque langue fournie.

Le répertoire suivant stocke également les fichiers de contrat de licence:

#### **/QIBM/ProdData/LicenseDoc/5724H72\_V8R0M0**

Fichiers de licence. Les fichiers sont nommés comme 5724H72\_V8R0M0\_ *xx* , où *xx* est l'identificateur de langue à 2 ou 5 caractères pour chaque langue fournie.

UserData:

### Présentation

#### QIBM

```
'-- UserData
    '-- mqm
    '-- errors
    '-- trace
    '-- qmgrs
```

```
'-- &system
'-- qmgrname1
'-- qmgrname2
'-- and so on
```

### **/QIBM/UserData/mqm**

Les sous-répertoires ci-dessous contiennent toutes les données utilisateur relatives aux gestionnaires de files d'attente.

Lorsque vous installez le produit, un fichier mqm.ini est créé dans le répertoire /QIBM/UserData/mqm/ (sauf s'il s'agit déjà d'une installation précédente).

Lorsque vous créez un gestionnaire de files d'attente, un fichier qm.ini est créé dans le répertoire /QIBM/UserData/mqm/qmgrs/ *QMGRNAME* / (où *QMGRNAME* est le nom du gestionnaire de files d'attente).

Les données des répertoires sont conservées lorsque le produit est supprimé.

## **Système de fichiers racine IFS pour IBM MQ MQI client**

Lorsque vous installez IBM MQ MQI client for IBM i, les répertoires suivants sont créés dans le système de fichiers racine IFS:

ProdData:

### **Présentation**

#### **QIBM**

```
'-- ProdData
'-- mqm
'-- lib
```

### **/QIBM/ProdData/mqm**

Les sous-répertoires situés sous ce répertoire contiennent toutes les données du produit. Les données de ce répertoire sont supprimées et remplacées chaque fois que le produit est remplacé.

UserData:

### **Présentation**

#### **QIBM**

```
'-- UserData
'-- mqm
'-- errors
'-- trace
```

### **/QIBM/UserData/mqm**

Les sous-répertoires situés sous ce répertoire contiennent toutes les données utilisateur.

## **Système de fichiers racine IFS pour IBM MQ Java**

Lorsque vous installez IBM MQ Java sous IBM i, les répertoires suivants sont créés dans le système de fichiers racine IFS:

ProdData:

### **Présentation**

#### **QIBM**

```
'-- ProdData
'-- mqm
'-- java
'-- samples
```

```
'-- bin
'-- lib
```

### **/QIBM/ProdData/mqm/java**

Les sous-répertoires ci-dessous contiennent toutes les données du produit, y compris les classes Java . Les données de ce répertoire sont supprimées et remplacées chaque fois que le produit est remplacé.

### **/QIBM/ProdData/mqm/java/samples**

Les sous-répertoires ci-dessous contiennent tous les exemples de classes et de données Java .

## **Bibliothèques créées par les installations serveur et client**

L'installation du serveur ou du client IBM MQ crée les bibliothèques suivantes:

- QMQM  
Bibliothèque de logiciel.
- QMQMSAMP  
La bibliothèque d'exemples (si vous choisissez d'installer les exemples).
- QMxxxx  
Serveur uniquement.

Chaque fois que vous créez un gestionnaire de files d'attente, IBM MQ crée automatiquement une bibliothèque associée, avec un nom tel que QMxxxx où xxxx est dérivé du nom du gestionnaire de files d'attente. Cette bibliothèque contient des objets spécifiques au gestionnaire de files d'attente, y compris les journaux et les récepteurs associés. Par défaut, le nom de cette bibliothèque est dérivé du nom du gestionnaire de files d'attente préfixé avec les caractères QM. Par exemple, pour un gestionnaire de files d'attente appelé TEST, la bibliothèque serait appelée QMTEST.

**Remarque :** Lorsque vous créez un gestionnaire de files d'attente, vous pouvez spécifier le nom de sa bibliothèque si vous le souhaitez. Exemple :

```
CRTMQM MQMNAME(TEST) MQMLIB(TESTLIB)
```

Vous pouvez utiliser la commande WRKLIB pour répertorier toutes les bibliothèques créées par IBM MQ for IBM i . Par rapport aux bibliothèques du gestionnaire de files d'attente, vous verrez le texte QMGR : QMGRNAME. La syntaxe de la commande est la suivante :

```
WRKLIB LIB(QM*)
```

Ces bibliothèques associées au gestionnaire de files d'attente sont conservées lorsque le produit est supprimé.

## **Multi Planification de la prise en charge du système de fichiers pour MFT sur Multiplatforms**

Les agents IBM MQ Managed File Transfer MFT peuvent être utilisés pour transférer des données vers et depuis des fichiers sur un système de fichiers. En outre, les moniteurs de ressources exécutés dans un agent peuvent être configurés pour surveiller les fichiers d'un système de fichiers.

MFT exige que ces fichiers soient stockés sur un système de fichiers prenant en charge le verrouillage. Il y a deux raisons à cela:

- Un agent verrouille un fichier pour s'assurer qu'il ne change pas une fois qu'il a commencé à lire des données à partir de celui-ci ou à y écrire des données.
- Les moniteurs de ressources verrouillent les fichiers pour vérifier qu'aucun autre processus ne les utilise actuellement.

Les agents et les moniteurs de ressources utilisent la méthode Java `FileChannel.tryLock()` pour effectuer le verrouillage et le système de fichiers doit pouvoir verrouiller les fichiers lorsqu'il lui est demandé de le faire à l'aide de cet appel.

**Important :** Les systèmes de fichiers suivants ne sont pas pris en charge car ils ne répondent pas aux exigences techniques d' MFT:

- GlusterFS
- NFS version 3

Multi

## Choix de la journalisation circulaire ou linéaire sur Multiplatforms

Dans IBM MQ, vous pouvez choisir la journalisation circulaire ou linéaire. Les informations suivantes vous donnent un aperçu des deux types.

### Avantages de la journalisation avec réutilisation automatique des journaux

Les principaux avantages de la journalisation avec réutilisation automatique des journaux sont les suivants:

- Plus facile à administrer.

Une fois que vous avez configuré correctement la journalisation avec réutilisation automatique des journaux pour votre charge de travail, aucune administration supplémentaire n'est nécessaire. En revanche, pour la journalisation linéaire, les images de support doivent être enregistrées et les extensions de journal qui ne sont plus nécessaires doivent être archivées ou supprimées.

- Meilleures performances

La journalisation avec réutilisation automatique des journaux est plus efficace que la journalisation linéaire, car la journalisation avec réutilisation automatique des journaux peut réutiliser des domaines qui ont déjà été formatés. Alors que la journalisation linéaire doit allouer de nouvelles extensions de journal et les formater.

Pour plus d'informations, voir [Gestion des journaux](#).

### Avantages de la journalisation linéaire

Le principal avantage de la journalisation linéaire est que la journalisation linéaire offre une protection contre un plus grand nombre de pannes.

Ni la journalisation linéaire ni la journalisation circulaire ne protègent contre un journal endommagé ou supprimé, ni les messages ou les files d'attente qui ont été supprimés par les applications ou l'administrateur.

La consignation linéaire (mais pas circulaire) permet de récupérer les objets endommagés. Ainsi, la consignation linéaire offre une protection contre les fichiers de file d'attente endommagés ou supprimés, car ces files d'attente endommagées peuvent être récupérées à partir d'un journal linéaire.

Protection circulaire et linéaire contre les pertes d'alimentation et les pannes de communication, comme décrit dans la rubrique [Récupération après une perte d'alimentation ou des pannes de communication](#).

### Autres considérations

Le choix de l'option linéaire ou circulaire dépend de la quantité de redondance dont vous avez besoin.

Il y a un coût à choisir pour plus de redondance, c'est-à-dire la journalisation linéaire, causée par le coût des performances et le coût d'administration.

Pour plus d'informations, voir [Types de journalisation](#).



## Mémoire partagée sur AIX

Si certains types d'application ne parviennent pas à se connecter en raison d'une limitation de mémoire AIX , dans la plupart des cas, cela peut être résolu en définissant la variable d'environnement EXTSHM=ON.

Certains processus 32 bits sous AIX peuvent rencontrer une limitation du système d'exploitation qui affecte leur capacité à se connecter aux gestionnaires de files d'attente IBM MQ . Chaque connexion standard à IBM MQ utilise de la mémoire partagée, mais contrairement aux autres plateformes UNIX , AIX autorise les processus 32 bits à n'associer que 11 ensembles de mémoire partagée.

La plupart des processus 32 bits ne rencontrent pas cette limite, mais les applications ayant des besoins de mémoire élevés risquent de ne pas se connecter à IBM MQ avec le code anomalie 2102: MQRC\_RESOURCE\_PROBLEM. Les types d'application suivants peuvent voir cette erreur:

- Programmes s'exécutant sur des machines virtuelles Java 32 bits
- Programmes utilisant les modèles de mémoire de grande taille ou de très grande taille
- Programmes se connectant à de nombreux gestionnaires de files d'attente ou bases de données
- Programmes qui se connectent à des ensembles de mémoire partagée par eux-mêmes

AIX offre une fonction de mémoire partagée étendue pour les processus 32 bits qui leur permet de connecter davantage de mémoire partagée. Pour exécuter une application avec cette fonction, exportez la variable d'environnement EXTSHM=ON avant de démarrer vos gestionnaires de files d'attente et votre programme. La fonction EXTSHM=ON permet d'éviter cette erreur dans la plupart des cas, mais elle est incompatible avec les programmes qui utilisent l'option SHM\_SIZE de la fonction shmctl.

Les applications IBM MQ MQI client et tous les processus 64 bits ne sont pas concernés par cette limitation. Ils peuvent se connecter à des gestionnaires de files d'attente IBM MQ que EXTSHM ait été défini ou non.

## Ressources IPC IBM MQ et UNIX System V

Un gestionnaire de files d'attente utilise certaines ressources IPC. Utilisez **ipcs -a** pour déterminer les ressources utilisées.

**Ces informations s'appliquent à IBM MQ exécuté sur des systèmes AIX and Linux uniquement.**

IBM MQ utilise les ressources de communication interprocessus (IPC) System V ( *sémaphores et segments de mémoire partagée* ) pour stocker et transmettre des données entre les composants système. Ces ressources sont utilisées par les processus et les applications du gestionnaire de files d'attente qui se connectent au gestionnaire de files d'attente. IBM MQ MQI clients n'utilise pas de ressources IPC, à l'exception du contrôle de trace IBM MQ . Utilisez la UNIX commande **ipcs -a** pour obtenir des informations complètes sur le nombre et la taille des ressources IPC actuellement utilisées sur la machine.

## Priorité des processus IBM MQ et UNIX

Bonnes pratiques lors de la définition des valeurs de priorité de processus *nice* .

**Ces informations s'appliquent à IBM MQ exécuté sur des systèmes AIX and Linux uniquement.**

Si vous exécutez un processus en arrière-plan, le shell appelant peut lui attribuer une valeur *nice* plus élevée (et donc une priorité plus faible). Cela peut avoir des implications générales sur les performances de IBM MQ . Dans les situations très stressées, s'il existe de nombreuses unités d'exécution prêtes à être exécutées avec une priorité plus élevée et certaines avec une priorité plus faible, les caractéristiques de planification du système d'exploitation peuvent priver les unités d'exécution de priorité plus faible du temps processeur.

Il est recommandé que les processus démarrés indépendamment associés aux gestionnaires de files d'attente, tels que **runmq1sr**, aient les mêmes valeurs *nice* que le gestionnaire de files d'attente auquel ils sont associés. Assurez-vous que le shell n'affecte pas une valeur *nice* plus élevée à ces processus d'arrière-plan. Par exemple, dans ksh, utilisez le paramètre "set +o bgnice" pour empêcher

ksh d'augmenter la valeur *nice* des processus d'arrière-plan. Vous pouvez vérifier les valeurs *nice* des processus en cours d'exécution en examinant la colonne *NI* d'une liste "ps -efl".

Démarrez également les processus d'application IBM MQ avec la même valeur *nice* que le gestionnaire de files d'attente. S'ils s'exécutent avec des valeurs *nice* différentes, une unité d'exécution d'application peut bloquer une unité d'exécution de gestionnaire de files d'attente, ou inversement, ce qui entraîne une dégradation des performances.

▶ z/OS

## Planning your IBM MQ environment on z/OS

When planning your IBM MQ environment, you must consider the resource requirements for data sets, page sets, Db2, Coupling Facilities, and the need for logging, and backup facilities. Use this topic to plan the environment where IBM MQ runs.

Before you plan your IBM MQ architecture, familiarize yourself with the basic IBM MQ for z/OS concepts, see the topics in [IBM MQ for z/OS concepts](#).

When planning your queue manager, you might need to work with different people in your organization. It is usually a good idea to involve those people early, as change control procedures can take a long time. They might also be able to tell you what parameters you need to configure IBM MQ for z/OS.

For example you might need to work with the:

- Storage administrator, to determine the high level qualifier of queue manager data sets, and to allocate enough space for queue manager data sets.
- z/OS system programmer to define the IBM MQ subsystem to z/OS and APF authorize the IBM MQ for z/OS libraries.
- Network administrator to determine which TCP/IP stack and ports should be used for IBM MQ for z/OS.
- Security administrator to set up access to queue manager data sets, security profiles for IBM MQ for z/OS resources, and TLS certificates.
- Db2 administrator to set up Db2 tables when configuring a queue sharing group.

### Related concepts

[IBM MQ Technical overview](#)

### Related tasks

[“Planification d'une architecture IBM MQ” on page 5](#)

Lors de la planification de l'environnement IBM MQ, prenez en compte la prise en charge fournie par IBM MQ pour les architectures à un ou plusieurs gestionnaires de files d'attente et pour les styles de messagerie de publication/abonnement et point à point. Prévoyez également vos exigences en matières de ressources et votre utilisation des fonctions de consignation et de sauvegarde.

[Configuring z/OS](#)

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

▶ z/OS

## Planning for your queue manager

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

The best way to configure a queue manager is in steps:

1. Configure the base queue manager
2. Configure the channel initiator which does queue manager to queue manager communications, and remote client application communication
3. If you want to encrypt and protect messages, configure [Advanced Message Security](#)
4. If you want to use File Transfer over IBM MQ, configure [Managed File Transfer for z/OS](#).
5. If you want to use the administrative or messaging REST API, or the IBM MQ Console to manage IBM MQ from a web browser, configure the mqweb server.

Some enterprises have hundreds of thousands of queue managers in their environment. You need to consider your IBM MQ network now, and in five years time.

On z/OS, some queue managers process thousands of messages a second, and log over 100 MB a second. If you expect very high volumes you may need to consider having more than one queue manager.

On z/OS, IBM MQ can run as part of a queue sharing group (QSG) where messages are stored in the Coupling Facility, and any queue manager in the queue sharing group can access the messages. If you want to run in a queue sharing group you need to consider how many queue managers you need. Typically, there is one queue manager for each LPAR. You might also have one queue manager to backup CF structures regularly.

Some changes to configuration are easy to do, such as defining a new queue. Some are harder, such as making logs and page sets bigger; and some configuration cannot be changed, such as the name of a queue manager or the queue sharing group name.

There is performance and tuning information available in the [MP16 performance SupportPac](#).

## Naming conventions

You need to have a naming convention for the queue manager data sets.

Many enterprises use the release number in the name of the load libraries, and so on. You might want to consider having an alias of MQM . SCSQAUTH pointing to the version currently in use, such as MQM . V930 . SCSQAUTH, so you do not have to change CICS®, Batch, and IMS JCL when you migrate to a new version of IBM MQ.

You can use a symbolic link in z/OS UNIX System Services to reference the installation directory for the version of IBM MQ currently in use.

The data sets used by the queue manager (logs, page sets, JCL libraries) need a naming convention to simplify the creation of security profiles, and the mapping of data sets to SMS storage classes that control where the data sets are placed on disk, and the attributes they have.

Note, that putting the version of IBM MQ into the name of the page sets or logs, is not a good idea. One day you might migrate to a new version, and the data set will have the "wrong" names.

## Applications

You need to understand the business applications and the best way to configure IBM MQ. For example if applications have logic to provide recovery and repeat capability, then non persistent messages might be good enough. If you want IBM MQ to handle the recovery, then you need to use persistent messages and put and get messages in syncpoint.

You need to isolate queues from different business transactions. If a queue for one business application fills up, you do not want this impacting other business applications. Isolate the queues in different page sets and buffer pools, or structures, if possible.

You need to understand the profile of messages. For many applications the queues have only a few messages. Other applications can have queues build up during the day, and be processed overnight. A queue which normally has only a few messages on it, might need to hold many hours worth of messages if there is a problem and messages are not processed. You need to size the CF structures and page sets to allow for your expected peak capacity.

## Post configuration

Once you have configured your queue manager (and components) you need to plan for:

- Backing up page sets.
- Backing up definitions of objects.
- Automating the backup of any CF structures.
- Monitoring IBM MQ messages, and taking action when a problem is detected.

- Collecting the IBM MQ statistics data.
- Monitoring resource usage, such as virtual storage, and amount of data logged per hour. With this you can see if your resource usage is increasing and if you need to take actions, such as setting up a new queue manager

## **Planning your storage and performance requirements on z/OS**

You must set realistic and achievable storage, and performance goals for your IBM MQ system. Use this topic help you understand the factors which affect storage, and performance.

This topic contains information about the storage and performance requirements for IBM MQ for z/OS. It contains the following sections:

- [z/OS performance options for IBM MQ](#)
- [Determining z/OS workload management importance and velocity goals](#)
- [“Library storage” on page 156](#)
- [“System LX usage” on page 157](#)
- [“Storage configuration” on page 158](#)
- [“Disk storage” on page 162](#)

See, [“Where to find more information about storage and performance requirements” on page 162](#) for more information.

### **z/OS performance options for IBM MQ**

With workload management, you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much resource, such as processor and storage, should be given to the work to meet its goal. Workload management controls the dispatching priority based on the goals you supply. Workload management raises or lowers the priority as needed to meet the specified goal. Thus, you need not fine-tune the exact priorities of every piece of work in the system and can focus instead on business objectives.

The three kinds of goals are:

#### **Response time**

How quickly you want the work to be processed

#### **Execution velocity**

How fast the work should be run when ready, without being delayed for processor, storage, I/O access, and queue delay

#### **Discretionary**

A category for low priority work for which there are no performance goals

Response time goals are appropriate for end-user applications. For example, CICS users might set workload goals as response time goals. For IBM MQ address spaces, velocity goals are more appropriate. A small amount of the work done in the queue manager is counted toward this velocity goal but this work is critical for performance. Most of the work done by the queue manager counts toward the performance goal of the end-user application. Most of the work done by the channel initiator address space counts toward its own velocity goal. The receiving and sending of IBM MQ messages, which the channel initiator accomplishes, is typically important for the performance of business applications using them.

### **Determining z/OS workload management importance and velocity goals**

See [“Determining z/OS workload management importance” on page 157](#) for more information.

### **Library storage**

You must allocate disk storage for the product libraries. The exact figures depend on your configuration, and should include both the target and distribution libraries, as well as the SMP/E libraries.

The target libraries used by IBM MQ for z/OS use PDSE formats. Ensure that any PDSE target libraries are not shared outside a sysplex. For more information about the required libraries and their sizes and the required format, see the Program Directory. Pour les liens de téléchargement des répertoires de programme, voir [IBM MQ for z/OS Program Directory PDF files](#).

## System LX usage

Each defined IBM MQ subsystem reserves one system linkage index (LX) at IPL time, and a number of non-system linkage indexes when the queue manager is started. The system linkage index is reused when the queue manager is stopped and restarted. Similarly, distributed queuing reserves one non-system linkage index. In the unlikely event of your z/OS system having inadequate system LXs defined, you might need to take these reserved system LXs into account.

If required, the number of system LXs can be increased by setting the *NSYSLX* parameter in SYS1.PARMLIB member IEASYSxx.

## **Determining z/OS workload management importance**

For full information about workload management and defining goals through the service definition, see the .z/OS product documentation.

This topic suggests how to set the z/OS workload management importance and velocity goals relative to other important work in your system. See [z/OS MVS Planning: Workload Management](#) for more information.

The queue manager address space needs to be defined with high priority as it provides subsystem services. The channel initiator is an application address space, but is usually given a high priority to ensure that messages being sent to a remote queue manager are not delayed. Advanced Message Security (AMS) also provides subsystem services and needs to be defined with high priority.

Use the following service classes:

### **The default SYSSTC service class**

- VTAM and TCP/IP address spaces
- IRLM address space (IRLMPROC)

**Note:** The VTAM, TCP/IP, and IRLM address spaces must have a higher dispatching priority than all the DBMS address spaces, their attached address spaces, and their subordinate address spaces. Do not allow workload management to reduce the priority of VTAM, TCP/IP, or IRLM to (or below) that of the other DBMS address spaces

### **A high velocity goal and importance of 1 for a service class with a name that you define, such as PRODREGN, for the following:**

- IBM MQ queue manager, channel initiator and AMS address spaces
- Db2 (all address spaces, except for the Db2-established stored procedures address space)
- CICS (all region types)
- IMS (all region types except BMPs)

A high velocity goal is good for ensuring that startups and restarts are performed as quickly as possible for all these address spaces.

The velocity goals for CICS and IMS regions are only important during startup or restart. After transactions begin running, workload management ignores the CICS or IMS velocity goals and assigns priorities based on the response time goals of the transactions that are running in the regions. These transaction goals should reflect the relative priority of the business applications they implement. They might typically have an importance value of 2. Any batch applications using IBM MQ should similarly have velocity goals and importance reflecting the relative priority of the business applications they implement. Typically the importance and velocity goals will be less than those for PRODREGN.

## z/OS Storage configuration

**V 9.4.0** In a 64 bit address space, there is a virtual line called "the bar" that marks the 2GB address. The bar separates storage below the 2GB address, called "below the bar", from storage above the 2GB address, called "above the bar". Storage below the bar uses 31 bit addressability, storage above the bar uses 64 bit addressability.

### **V 9.4.0**

You can specify the limit of 31-bit storage by using the JCL REGION parameter, and the limit of 64-bit storage by using the MEMLIMIT parameter. These specified values can be overridden by z/OS exits.

## Suggested storage configuration

The following table shows suggested **REGION** and **MEMLIMIT** values for the queue manager, channel initiator, and AMS address spaces. These suggestions should be used as a starting point and adjusted using the information in:

- “Queue manager storage configuration” on page 158
- “Channel initiator storage configuration from IBM MQ 9.4.0” on page 161

Address space	Storage configuration
Queue manager	REGION=0M, MEMLIMIT=3G
<b>V 9.4.0</b> Channel initiator from IBM MQ 9.4.0	REGION=0M, MEMLIMIT=2G
AMS address space	REGION=0M

## Managing the MEMLIMIT and REGION size

Other mechanisms, for example the **MEMLIMIT** parameter in the SMFPRMxx member of SYS1.PARMLIB or the IEFUSI exit might be used at your installation to provide a default amount of virtual storage above the bar for z/OS address spaces. See [Memory management above the bar](#) for full details about limiting storage above the bar.

### z/OS Queue manager storage configuration

The queue manager address space is likely to be the major user of 64-bit storage in an IBM MQ installation. Each connection to the queue manager requires common storage to be allocated as described in the following text. In addition to 64-bit storage, you should allow the queue manager to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

## Common storage

Each IBM MQ for z/OS subsystem has the following approximate storage requirements:

- CSA 4KB
- ECSA 800KB, plus the size of the trace table that is specified in the **TRACTBL** parameter of the CSQ6SYSP system parameter macro. For more information, see [Using CSQ6SYSP](#).

In addition, each concurrent logical connection to the queue manager requires about 5 KB of ECSA. When a task ends, other IBM MQ tasks can reuse this storage.

IBM MQ does not release the storage until the queue manager is shut down, so you can calculate the maximum amount of ECSA required by multiplying the maximum number of concurrent connections by 5KB. The number of concurrent logical connections is the sum of the number of:

- Tasks (TCBs) in Batch, TSO, z/OS UNIX System Services, IMS, and Db2 stored procedure address space (SPAS) regions that are connected to IBM MQ, but not disconnected.
- CICS transactions that have issued an IBM MQ request, but have not terminated
- JMS Connections, Sessions, TopicSessions or QueueSessions that have been created (for bindings connection), but not yet destroyed or garbage collected.
- Active IBM MQ channels

You can set a limit to the common storage, used by logical connections to the queue manager, with the **ACELIM** configuration parameter. The **ACELIM** control is primarily of interest to sites where Db2 stored procedures cause operations on IBM MQ queues.

When driven from a stored procedure, each IBM MQ operation can result in a new logical connection to the queue manager. Large Db2 units of work, for example due to table load, can result in an excessive demand for common storage.

**ACELIM** is intended to limit common storage use and to protect the z/OS system, by limiting the number of connections in the system. You should only set **ACELIM** on queue managers that have been identified as using excessive quantities of ECSA storage. See the **ACELIM** section in *Using CSQ6SYSP* for more information.

To set a value for **ACELIM**, firstly determine the amount of storage currently in the subpool controlled by the **ACELIM** value. This information is in the SMF 115 subtype 5 records produced by statistics CLASS(3) trace.

IBM MQ SMF data can be formatted using SupportPac MP1B. The number of bytes in use in the subpool controlled by **ACELIM** is displayed in the STGPPOOL DD, on the line titled *ACE/PEB*.

For more information about SMF 115 statistics records, see [Interpreting IBM MQ for z/OS performance statistics](#).

Increase the normal value by a sufficient margin to provide space for growth and workload spikes. Divide the new value by 1024 to yield a maximum storage size in KB for use in the **ACELIM** configuration.

## Private storage

The queue manager address space uses 64-bit storage for many internal control blocks. The **MEMLIMIT** parameter of the queue manager JCL defines the maximum amount of 64-bit storage available. 3GB of storage, **MEMLIMIT=3G**, is the minimum you should use, however, depending on your configuration significantly more might be required.

You should specify a specific **MEMLIMIT** value rather than **MEMLIMIT=NOLIMIT** to prevent potential problems. If you specify **NOLIMIT** or a very large value, then there is the potential to use up all of the available z/OS virtual storage, which leads to paging in your system. When increasing the value of **MEMLIMIT** you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for **MEMLIMIT** you might need to increase the size of your dump data sets as more data is captured in a dump.

You can monitor the address space storage usage from the **CSQY220I** message that indicates the amount of 31 and 64-bit private storage in use, and the remaining free amount.

## Buffer pools

Buffer pools are a significant user of private storage in the queue manager address space. Each buffer pool size is determined at queue manager initialization time, and storage is allocated for the buffer pool when a page set that is using that buffer pool is connected. The parameter **LOCATION (ABOVE|BELOW)** is used to specify where the buffers are allocated. You can use the **ALTER BUFFPOOL** command to dynamically change the size of buffer pools.

When calculating a value for **MEMLIMIT** it is critical that you take into account the buffer pool sizes if they are configured with **LOCATION (ABOVE)**. You should perform the calculation as follows.

Calculate the value of **MEMLIMIT** as 2GB plus the size of the buffer pools configured with **LOCATION (ABOVE)**, rounded up to the nearest GB. Set **MEMLIMIT** to a minimum of 3GB and increase this as necessary when you need to increase the size of your buffer pools.

For example, for three buffer pools configured with **LOCATION (ABOVE)**, buffer pool one has 10,000 buffers, and buffer pools two and three have 50,000 buffers each. Memory usage above the bar equals 110,000 (total number of buffers) \* 4096 = 450,560,000 bytes = 430MB.

All buffer pools regardless of **LOCATION** make use of 64-bit storage for control structures. As the number of buffer pools and number of buffers in those pools increase this can become significant. Each buffer requires around an additional 200 bytes of 64-bit storage. For the preceding configuration that would require: 200 \* 110,000 = 22,000,000 bytes = 21MB.

Therefore, in this scenario 3GB can be used for the **MEMLIMIT**, which allows scope for growth: 21MB + 430MB + 2GB which rounds up to 3GB.

For some configurations there can be significant performance benefits to using buffer pools that have their buffers permanently backed by real storage. You can achieve this by specifying the **FIXED4KB** value for the **PAGECLAS** attribute of the buffer pool. However, you should only do this if there is sufficient real storage available on the LPAR, otherwise other address spaces might be affected. For information about when you should use the **FIXED4KB** value for **PAGECLAS**, see IBM MQ Support Pac [MP16: IBM MQ for z/OS - Capacity planning & tuning](#).

Making the buffer pools so large that there is MVS™ paging might adversely affect performance. You might consider using a smaller buffer pool that does not page, with IBM MQ moving the message to and from the page set.

## Indexed queues

On z/OS, local queues are indexed if the queue has an **INDXTYPE** attribute that has not been set to **NONE**. The indexes for shared queues are held in a coupling facility, but for private queues the index is held in 64 bit storage. For each message on an indexed queue 136 bytes of data are used to index the message. For very deep queues this can result in a significant amount of 64 bit storage being allocated. For example, 10 million messages on an indexed queue will use 1.27 GB of 64 bit storage in order to maintain the index.

If you expect to have a large number of messages on indexed queues you should allow for this when setting **MEMLIMIT**. To calculate an upper limit for the amount of storage required for indexes, multiply the **MAXDEPTH** attribute for each indexed queue by 136 and sum the value. This value should be added to your existing **MEMLIMIT**.

## V 9.4.0 RECOVER CFSTRUCT

From IBM MQ 9.4.0 the **RECOVER CFSTRUCT** command makes greater use of 64-bit storage. In many cases there should be spare 64-bit storage available and so use of the command does not require an increase in the value of **MEMLIMIT**. However, if you are likely to have large structure backups, containing more than a few million messages, you should increase the **MEMLIMIT** for all queue managers which might process the **RECOVER CFSTRUCT** command by 500MB.

For example if you had **MEMLIMIT=3G** already, you should consider using **MEMLIMIT=4G** as the **MEMLIMIT** parameter does not allow for decimal points.

## Shared Message Data Set (SMDS) buffers and MEMLIMIT

When running messaging workloads using shared message data sets, there are two levels of optimizations that can be achieved by adjusting the **DSBUFS** and **DSBLOCK** attributes.

The amount of above bar queue manager storage used by the SMDS buffer is **DSBUFS x DSBLOCK**. This means that by default, 100 x 256KB (25MB) is used for each **CFLEVEL(5)** structure in the queue manager.

Although this value is not too high, if your enterprise, or enterprises have many **CFSTRUCTS**, some of them might allocate a high value of **MEMLIMIT** for buffer pools, and sometimes they have deep indexed queues, so in total, they might run out of storage above the bar.



The channel initiator typically uses much less 64-bit storage than the queue manager. However, from IBM MQ 9.4.0 the usage has increased. In addition to 64-bit storage, you should allow the channel initiator to use all available 31-bit storage by specifying REGION=0M on the queue manager JCL.

## Common storage

The channel initiator typically requires ECSA usage of up to 160KB.

## 31-bit private storage

The 31-bit storage available to the channel initiator limits the number of concurrent connections the CHINIT can have.

Every channel uses approximately 170KB of extended private region in the channel initiator address space. For message channels, for example, sender or receiver channels, storage is increased by message size if messages larger than 32KB are transmitted. This increased storage is freed when:

- A sending or client channel requires less than half the current buffer size for 10 consecutive messages.
- A heartbeat is sent or received.

The storage is freed for reuse within the Language Environment, however, the storage is not seen as free by the z/OS virtual storage manager. This means that the upper limit for the number of channels is dependent on message size and arrival patterns, and on limitations of individual user systems on extended private region size.

The upper limit on the number of channels is likely to be approximately 9000 on many systems because the extended region size is unlikely to exceed 1.6GB.

The channel initiator trace is written to a data space. The size of the data space storage, is controlled by the **TRAXTBL** parameter. See [ALTER QMGR](#).

## 64-bit private storage

The MEMLIMIT parameter of the channel initiator JCL defines the maximum amount of 64-bit storage available. 2 GB of storage, MEMLIMIT=2 GB, is the minimum value you should use. Depending on your configuration significantly more might be required.

You should specify a sensible MEMLIMIT value rather than MEMLIMIT=NOLIMIT to prevent potential problems. If you specify NOLIMIT or a very large value, then there is the potential to use up all of the available z/OS virtual storage, leading to paging in your system. When increasing the value of MEMLIMIT you should discuss the new setting with your z/OS system programmer in case there is a system-wide limit on the amount of on storage that can be used.

If you have a large value for MEMLIMIT you might need to increase the size of your dump data sets as more data is captured in a dump.

There are two users of 64-bit storage in the channel initiator: SMF and server-connection channels.

### SMF

If enabled, SMF class 4 accounting, or statistics, require 64-bit storage. A minimum of 256MB storage is required. If sufficient storage is not available, the channel initiator issues the [CSQX124E](#) message and class 4 accounting and statistics are not available.

### Server-connection channels

From IBM MQ 9.4.0 server-connection channels allocate message buffers in 64-bit storage, if they are transferring messages larger than 32 KB in size.

These buffers are freed if the channels require less than half the current buffer size for 10 consecutive messages, or a heartbeat is sent or received.

The value of MEMLIMIT sets an upper limit on how many concurrent server-connection channels can run. You should use a minimum value of MEMLIMIT=2G to ensure that the same number of channels can run as in earlier versions of IBM MQ, as well as providing some capacity for growth.

You can calculate an approximate value for MEMLIMIT by working out the peak maximum number of concurrently active server-connection channels, and for those channels the maximum message size you expect them to transfer. You should use MEMLIMIT=2GB as a starting point and round up.

For example, if you set the maximum number of concurrent server-connection channels to be 2,000 and each channel to have a maximum message size of 1MB, then server-connection channels are using a maximum of just under 2GB of 64-bit storage. As this is very close to 2GB then you should round up to MEMLIMIT=3G.

## **Disk storage**

Use this topic when planning your disk storage requirements for log data sets, Db2 storage, coupling facility storage, and page data sets.

Work with your storage administrator to determine where to put the queue manager data sets. For example, your storage administrator may give you specific DASD volumes, or SMS storage classes, data classes, and management classes for the different data set types.

- Log data sets must be on DASD. These logs can have high I/O activity with a small response time and do not need to be backed up.
- Archive logs can be on DASD or tape. After they have been created, they might never be read again except in an abnormal situation, such as recovering a page set from a backup. They should have a long retention date.
- Page sets might have low to medium activity and should be backed up regularly. On a high use system, they should be backed up twice a day.
- BSDS data sets should be backed up daily; they do not have high I/O activity.

All data sets are similar to those used by Db2, and similar maintenance procedures can be used for IBM MQ.

See the following sections for details of how to plan your data storage:

- **Logs and archive storage**

[“How long do I need to keep archive logs” on page 181](#) describes how to determine how much storage your active log and archive data sets require, depending on the volume of messages that your IBM MQ system handles and how often the active logs are offloaded to your archive data sets.

- **Db2 storage**

[“Db2 storage” on page 198](#) describes how to determine how much storage Db2 requires for the IBM MQ data.

- **coupling facility storage**

[“Defining coupling facility resources” on page 188](#) describes how to determine how large to make your coupling facility structures.

- **Page set and message storage**

[“Planning your page sets and buffer pools” on page 163](#) describes how to determine how much storage your page data sets require, depending on the sizes of the messages that your applications exchange, on the numbers of these messages, and on the rate at which they are created or exchanged.

## **Where to find more information about storage and performance requirements**

Use this topic as a reference to find more information about storage and performance requirements.

You can find more information from the following sources:

<i>Table 20. Where to find more information about storage requirements</i>	
<b>Topic</b>	<b>Where to look</b>
System parameters	<a href="#">Using CSQ6SYSP</a> and <a href="#">Customizing your queue managers</a>
Storage required to install IBM MQ	Program Directory. Pour les liens de téléchargement des répertoires de programme, voir <a href="#">IBM MQ for z/OS Program Directory PDF files</a> .
IEALIMIT and IEFUSI exits	See <a href="#">IEALIMIT</a> and <a href="#">IEFUSI</a> in the <i>z/OS:MVS Installation Exits</i> documentation.
Latest information	IBM MQ SupportPac website <a href="#">SupportPacs pour IBM MQ et autres zones de projet</a> .
Workload management and defining goals through the service definition	<a href="#">z/OS MVS Planning: Workload Management</a>

## **Planning your page sets and buffer pools**

Information to help you with planning the initial number, and sizes of your page data sets, and buffer pools.

This topic contains the following sections:

- [“Plan your page sets” on page 163](#)
  - [Page set usage](#)
  - [Number of page sets](#)
  - [Size of page sets](#)
  - [Planning for z/OS data set encryption](#)
- [“Calculate the size of your page sets” on page 164](#)
  - [Page set zero](#)
  - [Page set 01 - 99](#)
  - [Calculating the storage requirement for messages](#)
- [“Enabling dynamic page set expansion” on page 166](#)
- [“Defining your buffer pools” on page 168](#)

### **Plan your page sets**

#### **Page set usage**

For short-lived messages, few pages are normally used on the page set and there is little or no I/O to the data sets except at startup, during a checkpoint, or at shutdown.

For long-lived messages, those pages containing messages are normally written out to disk. This operation is performed by the queue manager in order to reduce restart time.

Separate short-lived messages from long-lived messages by placing them on different page sets and in different buffer pools.

#### **Number of page sets**

Using several large page sets can make the role of the IBM MQ administrator easier because it means that you need fewer page sets, making the mapping of queues to page sets simpler.

Using multiple, smaller page sets has a number of advantages. For example, they take less time to back up, and I/O can be carried out in parallel during backup and restart. However, consider that this adds a significant performance cost to the role of the IBM MQ administrator, who is required to map each queue to one of a much greater number of page sets.

Define at least five page sets, as follows:

- A page set reserved for object definitions (page set zero)
- A page set for system-related messages
- A page set for performance-critical long-lived messages
- A page set for performance-critical short-lived messages
- A page set for all other messages

[“Defining your buffer pools” on page 168](#) explains the performance advantages of distributing your messages on page sets in this way.

### Size of page sets

Define sufficient space in your page sets for the expected peak message capacity. Consider for any unexpected peak capacity, such as when a build-up of messages develops because a queue server program is not running. You can do this by allocating the page set with secondary extents or, alternatively, by enabling dynamic page set expansion. For more information, see [“Enabling dynamic page set expansion” on page 166](#). It is difficult to make a page set smaller, so it is often better to allocate a smaller page set, and allow it to expand when needed.

When planning page set sizes, consider all messages that might be generated, including non-application message data. For example, trigger messages, event messages and any report messages that your application has requested.

The size of the page set determines the time taken to recover a page set when restoring from a backup, because a large page set takes longer to restore.

**Note:** Recovery of a page set also depends on the time the queue manager takes to process the log records written since the backup was taken; this time period is determined by the backup frequency. For more information, see [“Planning for backup and recovery” on page 200](#).

**Note:** Page sets larger than 4 GB require the use of SMS extended addressability.

### Planning for z/OS data set encryption

You can apply the z/OS data set encryption feature to page sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these page sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, [confidentiality for data at rest on IBM MQ for z/OS with data set encryption](#), for more information.

### Calculate the size of your page sets

For queue manager object definitions (for example, queues and processes), it is simple to calculate the storage requirement because these objects are of fixed size and are permanent. For messages however, the calculation is more complex for the following reasons:

- Messages vary in size.
- Messages are transitory.
- Space occupied by messages that have been retrieved is reclaimed periodically by an asynchronous process.

Large page sets of greater than 4 GB that provide extra capacity for messages if the network stops, can be created if required. It is not possible to modify the existing page sets. Instead, new page sets with extended addressability and extended format attributes, must be created. The new page sets must be

the same physical size as the old ones, and the old page sets must then be copied to the new ones. If backward migration is required, page set zero must not be changed. If page sets less than 4 GB are adequate, no action is needed.

### Page set zero

Page set zero is reserved for object definitions.

For page set zero, the storage required is:

```
(maximum number of local queue definitions x 1010)
(excluding shared queues)
+ (maximum number of model queue definitions x 746)
+ (maximum number of alias queue definitions x 338)
+ (maximum number of remote queue definitions x 434)
+ (maximum number of permanent dynamic queue definitions x 1010)
+ (maximum number of process definitions x 674)
+ (maximum number of namelist definitions x 12320)
+ (maximum number of message channel definitions x 2026)
+ (maximum number of client-connection channel definitions x 5170)
+ (maximum number of server-connection channel definitions x 2026)
+ (maximum number of storage class definitions x 266)
+ (maximum number of authentication information definitions x 1010)
+ (maximum number of administrative topic definitions x 15000)
+ (total length of topic strings defined in administrative topic definitions)
```

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

You do not need to allow for objects that are stored in the shared repository, but you must allow for objects that are stored or copied to page set zero (objects with a disposition of GROUP or QMGR).

The total number of objects that you can create is limited by the capacity of page set zero. The number of local queues that you can define is limited to 524 287.

### Page sets 01 - 99

For page sets 01 - 99, the storage required for each page set is determined by the number and size of the messages stored on that page set. (Messages on shared queues are not stored on page sets.)

Divide this value by 4096 to determine the number of records to specify in the cluster for the page set data set.

### Calculating the storage requirement for messages

This section describes how messages are stored on pages. Understanding this can help you calculate how much page set storage you must define for your messages. To calculate the approximate space required for all messages on a page set you must consider maximum queue depth of all the queues that map to the page set and the average size of messages on those queues.

**Note:** The sizes of the structures and control information given in this section are liable to change between major releases. For details specific to your release of IBM MQ, refer to [SupportPac MP16 - IBM MQ for z/OS Planification et optimisation de la capacité](#) and [Famille IBM MQ -Rapports de performances](#)

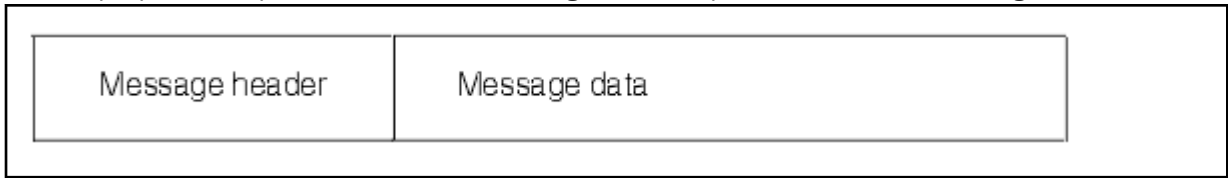
You must allow for the possibility that message "gets" might be delayed for reasons outside the control of IBM MQ (for example, because of a problem with your communications protocol). In this case, the "put" rate of messages might far exceed the "get" rate. This can lead to a large increase in the number of messages stored in the page sets and a consequent increase in the storage size demanded.

Each page in the page set is 4096 bytes long. Allowing for fixed header information, each page has 4057 bytes of space available for storing messages.

When calculating the space required for each message, the first thing you must consider is whether the message fits on one page (a short message) or whether it needs to be split over two or more

pages (a long message). When messages are split in this way, you must allow for additional control information in your space calculations.

For the purposes of space calculation, a message can be represented as the following:



The message header section contains the message descriptor and other control information, the size of which varies depending on the size of the message. The message data section contains all the actual message data, and any other headers (for example, the transmission header or the IMS bridge header).

A minimum of two pages are required for page set control information which, is typically less than 1% of the total space required for messages.

### Short messages

A short message is defined as a message that fits on one page.

Small messages are stored one on each page.

### Long messages

If the size of the message data is greater than 3596 bytes, but not greater than 4 MB, the message is classed as a long message. When presented with a long message, IBM MQ stores the message on a series of pages, and stores control information that points to these pages in the same way that it would store a short message. This is shown in Figure 41 on page 166:

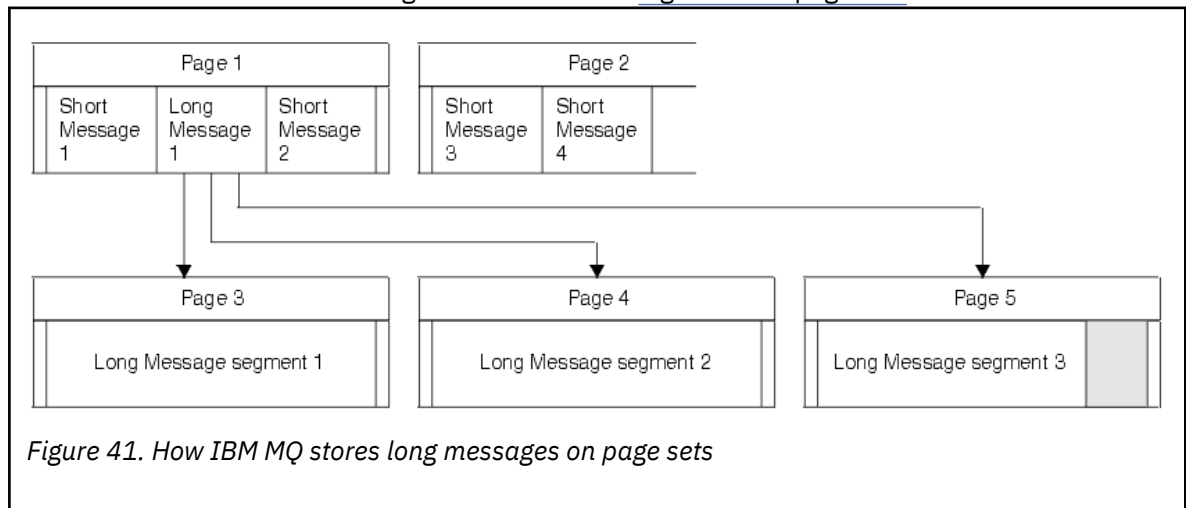


Figure 41. How IBM MQ stores long messages on page sets

### Very long messages

Very long messages are messages with a size greater than 4 MB. These are stored so that each 4 MB uses 1037 pages. Any remainder is stored in the same way as a long message, as described above.

### Enabling dynamic page set expansion

Page sets can be extended dynamically while the queue manager is running. A page set can have 123 extents, and can be spread over multiple disk volumes.

Each time a page set expands, a new data set extent is used. The queue manager continues to expand a page set when required, until the maximum number of extents has been reached, or until no more storage is available for allocation on eligible volumes.

Once page set expansion fails for one of the reasons above, the queue manager marks the page set for no further expansion attempts. This marking can be reset by altering the page set to EXPAND(SYSTEM).

Page set expansion takes place asynchronously to all other page set activity, when 90% of the existing space in the page set is allocated.

The page set expansion process formats the newly allocated extent and makes it available for use by the queue manager. However, none of the space is available for use, until the entire extent has been formatted. This means that expansion by a large extent is likely to take some time, and putting applications might 'block' if they fill the remaining 10% of the page set before the expansion has completed.

Sample thlqual.SCSQPROC(CSQ4PAGE) shows how to define the secondary extents.

To control the size of new extents, you use one of the following options of the EXPAND keyword of the DEFINE PSID and ALTER PSID commands:

- USER
- SYSTEM
- NONE

#### **USER**

Uses the secondary extent size specified when the page set was allocated. If a value was not specified, or if a value of zero was specified, dynamic page set expansion cannot occur.

Page set expansion occurs when the space in the page is 90% used, and is performed asynchronously with other page set activity.

This may lead to expansion by more than a single extent at a time.

Consider the following example: you allocate a page set with a primary extent of 100,000 pages and a secondary extent of 5000 pages. A message is put that requires 9999 pages. If the page set is already using 85,000 pages, writing the message crosses the 90% full boundary (90,000 pages). At this point, a further secondary extent is allocated to the primary extent of 100,000 pages, taking the page set size to 105,000 pages. The remaining 4999 pages of the message continue to be written. When the used page space reaches 94,500 pages, which is 90% of the updated page set size of 105,000 pages, another 5000 page extent is allocated, taking the page set size to 110,000 pages. At the end of the MQPUT, the page set has expanded twice, and 94,500 pages are used. None of the pages in the second page set expansion have been used, although they were allocated.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set. Only one extent is required to reach this size.

#### **SYSTEM**

Ignores the secondary extent size that was specified when the page set was defined. Instead, the queue manager sets a value that is approximately 10% of the current page set size. The value is rounded up to the nearest cylinder of DASD.

If a value was not specified, or if a value of zero was specified, dynamic page set expansion can still occur. The queue manager sets a value that is approximately 10% of the current page set size. The new value is rounded up depending on the characteristics of the DASD.

Page set expansion occurs when the space in the page set is approximately 90% used, and is performed asynchronously with other page set activity.

At restart, if a previously used page set has been replaced with a data set that is smaller, it is expanded until it reaches the size of the previously used data set.

#### **NONE**

No further page set expansion is to take place.

#### **Related reference**

[ALTER PSID](#)

[DEFINE PSID](#)

## ***Defining your buffer pools***

Use this topic to help plan the number of buffer pools you should define, and their settings.

This topic is divided into the following sections:

1. [“Decide on the number of buffer pools to define” on page 168](#)
2. [“Decide on the settings for each buffer pool” on page 169](#)
3. [“Monitor the performance of buffer pools under expected load” on page 169](#)
4. [“Adjust buffer pool characteristics” on page 169](#)

## **Decide on the number of buffer pools to define**

You should define four buffer pools initially:

### **Buffer pool 0**

Use for object definitions (in page set zero) and performance critical, system related message queues, such as the SYSTEM.CHANNEL.SYNCQ queue and the SYSTEM.CLUSTER.COMMAND.QUEUE and SYSTEM.CLUSTER.REPOSITORY.QUEUE queues.

However it is important to consider point [“7” on page 170](#) in *Adjust buffer pool characteristics* if a large number of channels, or clustering, is to be used.

Use the remaining three buffer pools for user messages.

### **Buffer pool 1**

Use for important long-lived messages.

Long-lived messages are those that remain in the system for longer than two checkpoints, at which time they are written out to the page set. If you have many long-lived messages, this buffer pool should be relatively small, so that page set I/O is evenly distributed (older messages are written out to DASD each time the buffer pool becomes 85% full).

If the buffer pool is too large, and the buffer pool never gets to 85% full, page set I/O is delayed until checkpoint processing. This might affect response times throughout the system.

If you expect few long-lived messages only, define this buffer pool so that it is sufficiently large to hold all these messages.

### **Buffer pool 2**

Use for performance-critical, short-lived messages.

There is normally a high degree of buffer reuse, using few buffers. However, you should make this buffer pool large to allow for unexpected message accumulation, for example, when a server application fails.

### **Buffer pool 3**

Use for all other (typically, performance noncritical) messages.

Queues such as the dead-letter queue, SYSTEM.COMMAND.\* queues and SYSTEM.ADMIN.\* queues can also be mapped to buffer pool 3.

Where virtual storage constraints exist, and buffer pools need to be smaller, buffer pool 3 is the first candidate for size reduction.

You might need to define additional buffer pools in the following circumstances:

- If a particular queue is known to require isolation, perhaps because it exhibits different behavior at various times.
  - Such a queue might either require the best performance possible under the varying circumstances, or need to be isolated so that it does not adversely affect the other queues in a buffer pool.
  - Each such queue can be isolated into its own buffer pool and page set.



- You want to isolate different sets of queues from each other for class-of-service reasons.
  - Each set of queues might then require one, or both, of the two types of buffer pools 1 or 2, as described in [Suggested definitions for buffer pool settings](#), necessitating creation of several buffer pools of a specific type.

## Decide on the settings for each buffer pool

If you are using the four buffer pools described in “[Decide on the number of buffer pools to define](#)” on page 168, then [Suggested definitions for buffer pool settings](#) gives two sets of values for the size of the buffer pools.

The first set is suitable for a test system, the other for a production system or a system that will become a production system eventually. In all cases define your buffer pools with the **LOCATION(ABOVE)** attribute

Definition setting	Test system	Production system
BUFFPOOL 0	1 050 buffers	50 000 buffers
BUFFPOOL 1	1 050 buffers	20 000 buffers
BUFFPOOL 2	1 050 buffers	50 000 buffers
BUFFPOOL 3	1 050 buffers	20 000 buffers

If you need more than the four suggested buffer pools, select the buffer pool (1 or 2) that most accurately describes the expected behavior of the queues in the buffer pool, and size it using the information in [Suggested definitions for buffer pool settings](#).

Ensure that your MEMLIMIT is set high enough, so that all the buffer pools can be located above the bar.

## Monitor the performance of buffer pools under expected load

You can monitor the usage of buffer pools by analyzing buffer pool performance statistics. In particular, you should ensure that the buffer pools are large enough so that the values of QPSTSOS, QPSTSTLA, and QPSTDMC remain at zero.

For further information, see [Buffer manager data records](#).

## Adjust buffer pool characteristics

Use the following points to adjust the buffer pool settings from “[Decide on the settings for each buffer pool](#)” on page 169, if required.

Use the performance statistics from “[Monitor the performance of buffer pools under expected load](#)” on page 169 as guidance.

1. If you are migrating from an earlier version of IBM MQ, only change your existing settings if you have more real storage available.
2. In general, bigger buffer pools are better for performance, and buffer pools can be much bigger if they are above the bar.

However, at all times you should have sufficient real storage available so that the buffer pools are resident in real storage. It is better to have smaller buffer pools that do not result in paging, than big ones that do.

Additionally, there is no point having a buffer pool that is bigger than the total size of the page sets that use it, although you should take into account page set expansion if it is likely to occur.

3. Aim for one page set per buffer pool, as this provides better application isolation.
4. If you have sufficient real storage, such that your buffer pools will never be paged out by the operating system, consider using page-fixed buffers in your buffer pool.

This is particularly important if the buffer pool is likely to undergo much I/O, as it saves the CPU cost associated with page-fixing the buffers before the I/O, and page-unfixing them afterwards.

5. There are several benefits to locating buffer pools above the bar even if they are small enough to fit below the bar. These are:
  - 31 bit virtual storage constraint relief - for example more space for common storage.
  - If the size of a buffer pool needs to be increased unexpectedly while it is being heavily used, there is less impact and risk to the queue manager, and its workload, by adding more buffers to a buffer pool that is already above the bar, than moving the buffer pool to above the bar and then adding more buffers.
6. Tune buffer pool zero and the buffer pool for short-lived messages (buffer pool 2) so that the 15% free threshold is never exceeded (that is, QPSTCBSL divided by QPSTNBUF is always greater than 15%). If more than 15% of buffers remain free, I/O to the page sets using these buffer pools can be largely avoided during normal operation, although messages older than two checkpoints are written to page sets.



**Attention:** The optimum value for these parameters is dependent on the characteristics of the individual system. The values given are intended only as a guideline and might not be appropriate for your system.

7. SYSTEM.\* queues which get very deep, for example SYSTEM.CHANNEL.SYNCQ, might benefit from being placed in their own buffer pool, if sufficient storage is available.

IBM MQ SupportPac MP16 - [IBM MQ for z/OS Planification et optimisation de la capacité](#) provides further information about tuning buffer pools.

## **Planning your logging environment**

Use this topic to plan the number, size and placement of the logs, and log archives used by IBM MQ.

Logs are used to:

- Write recovery information about persistent messages
- Record information about units of work using persistent messages
- Record information about changes to objects, such as define queue
- Backup CF structures

and for other internal information.

The IBM MQ logging environment is established using the system parameter macros to specify options, such as: whether to have single or dual active logs, what media to use for the archive log volumes, and how many log buffers to have.

These macros are described in [Create the bootstrap and log data sets](#) and [Tailor your system parameter module](#).

**Note:** If you are using queue sharing groups, ensure that you define the bootstrap and log data sets with SHAREOPTIONS(2 3).

This section contains information about the following topics:

### **Log data set definitions**

Use this topic to decide on the most appropriate configuration for your log data sets.

This topic contains information to help you answer the following questions:

- [Should your installation use single or dual logging?](#)
- [How many active log data sets do you need?](#)
- [“How large should the active logs be?” on page 172](#)
- [Active log placement](#)

- [“Active log encryption with z/OS data set encryption” on page 173](#)

## Should your installation use single or dual logging?

In general you should use dual logging for production, to minimize the risk of losing data. If you want your test system to reflect production, both should use dual logging, otherwise your test systems can use single logging.

With single logging data is written to one set of log data sets. With dual logging data is written to two sets of log data sets, so in the event of a problem with one log data set, such as the data set being accidentally deleted, the equivalent data set in the other set of logs can be used to recover the data.

With dual logging you require twice as much DASD as with single logging.

If you are using dual logging, then also use dual BSDSs and dual archiving to ensure adequate provision for data recovery.

Dual active logging adds a small performance cost.



**Attention:** Use of disk mirroring technologies, such as Metro Mirror, are not necessarily a replacement for dual logging and dual BSDS. If a mirrored data set is accidentally deleted, both copies are lost.

If you use persistent messages, single logging can increase maximum capacity by 10-30% and can also improve response times.

Single logging uses 2 - 310 active log data sets, whereas dual logging uses 4 - 620 active log data sets to provide the same number of active logs. Thus single logging reduces the amount of data logged, which might be important if your installation is I/O constrained.

## How many active log data sets do you need?

The number of logs depends on the activities of your queue manager. For a test system with low throughput, three active log data sets might be suitable. For a high throughput production system you might want the maximum number of logs available, so, if there is a problem with offloading logs you have more time to resolve the problems.

You must have at least three active log data sets, but it is preferable to define more. For example, if the time taken to fill a log is likely to approach the time taken to archive a log during peak load, define more logs.

**Note:** Page sets and active log data sets are eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV) and an archive log dataset can also reside in the EAS.

You should also define more logs to offset possible delays in log archiving. If you use archive logs on tape, allow for the time required to mount the tape.

Consider having enough active log space to keep a day's worth of data, in case the system is unable to archive because of lack of DASD or because it cannot write to tape. If all the active logs fill up, then IBM MQ is unable to process persistent messages or transactions. It is very important to have enough active log space.

It is possible to dynamically define new active log data sets as a way of minimizing the effect of archive delays or problems. New data sets can be brought online rapidly, using the **DEFINE LOG** command to avoid queue manager 'stall' due to lack of space in the active log.

If you want to define more than 31 active log data sets, you must configure your logging environment to use a version 2 format BSDS. Once a version 2 format BSDS is in use, up to 310 active log data sets can be defined for each log copy ring. See [“Planning to increase the maximum addressable log range” on page 182](#) for information on how you convert to a version 2 format BSDS.

You can tell whether your queue manager is using a version 2 or higher BSDS, either by running the print log map utility ([CSQJU004](#)), or from the [CSQJ034I](#) message issued during queue manager initialization.

An end of log RBA range of FFFFFFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 2, or higher, format BSDS is in use. An end of log RBA range of 0000FFFFFFFFFFFF, in the CSQJ034I message, indicates that a version 1 format BSDS is in use.

When a queue manager is using a version 2, or higher, format BSDS, it is possible to use the **DEFINE LOG** command to dynamically add more than 31 active log data sets to a log copy ring.

## How large should the active logs be?

The maximum supported active log size, when archiving to disk or to tape, is 4 GB.

You should create active logs of at least 1 GB in size for production and test systems.

**Important:** You need to be careful when allocating data sets, because IDCAMS rounds up the size you allocate.

To allocate a 3 GB log specify one of the following options:

- Cylinders(4369)
- Megabytes(3071)
- TRACKS(65535)
- RECORD(786420)

Any one of these allocates 2.99995 GB.

To allocate a 4GB log specify one of the following options:

- Cylinders(5825)
- Megabytes(4095)
- TRACKS(87375)
- RECORD(1048500)

Any one of these allocates 3.9997 GB.

When using striped data sets, where the data set is spread across multiple volumes, the specified size value is allocated on each DASD volume used for striping. So, if you want to use 4 GB logs and four volumes for striping, you should specify:

- CYLinders(1456)
- Megabytes(1023)

Setting these attributes allocates  $4 * 1456 = 5824$  Cylinders or  $4 * 1023 = 4092$  Megabytes.

**Note:** Striping is supported when using extended format data sets. This is usually set by the storage manager.

See [Increasing the size of the active log](#) for information on carrying out the procedure.

## Active log placement

You should work with your storage management team to set up storage pools for the queue managers. You need to consider:

- A naming convention, so the queue managers use the correct SMS definitions.
- Space required for active and archive logs. Your storage pool should have enough space for the active logs from a whole day.
- Performance and resilience to failures.

For performance reasons you should consider striping your active log data sets. The I/O is spread across multiple volumes and reduces the I/O response times, leading to higher throughput. See the preceding text for information about allocating the size of the active logs when using striping.

You should review the I/O statistics using reports from RMF or a similar product. Perform the review of these statistics monthly (or more frequently) for the IBM MQ data sets, to ensure there are no delays due to the location of the data sets.

In some situations, there can be much IBM MQ page set I/O, and this can impact the IBM MQ log performance if they are located on the same DASD.

If you use dual logging, ensure that each set of active and archive logs is kept apart. For example, allocate them on separate DASD subsystems, or on different devices.

This reduces the risk of them both being lost if one of the volumes is corrupted or destroyed. If both copies of the log are lost, the probability of data loss is high.

When you create a new active log data, set you should preformat it using `CSQJUFMT`. If the log is not preformatted, the queue manager formats the log the first time it is used, which impacts the performance.

With older DASD with large spinning disks, you had to be careful which volumes were used to get the best performance.

With modern DASD, where data is spread over many PC sized disks, you do not need to worry so much about which volumes are used.

Your storage manager should be checking the enterprise DASD to review and resolve any performance problems. For availability, you might want to use one set of logs on one DASD subsystem, and the dual logs on a different DASD subsystem.

## Active log encryption with z/OS data set encryption

You can apply the z/OS data set encryption feature to active log data sets for queue managers running at IBM MQ for z/OS 9.1.4 or later.

You must allocate these active log data sets with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

See the section, [confidentiality for data at rest on IBM MQ for z/OS with data set encryption](#), for more information.

### **Using MetroMirror with IBM MQ**

IBM Metro Mirror, previously known as Synchronous Peer to Peer Remote Copy (PPRC), is a synchronous replication solution between two storage subsystems, where write operations are completed on both the primary and secondary volumes before the write operation is considered to be complete. Metro Mirror can be used in environments that require no data loss in the event of a storage subsystem failure.

## Supported data set types

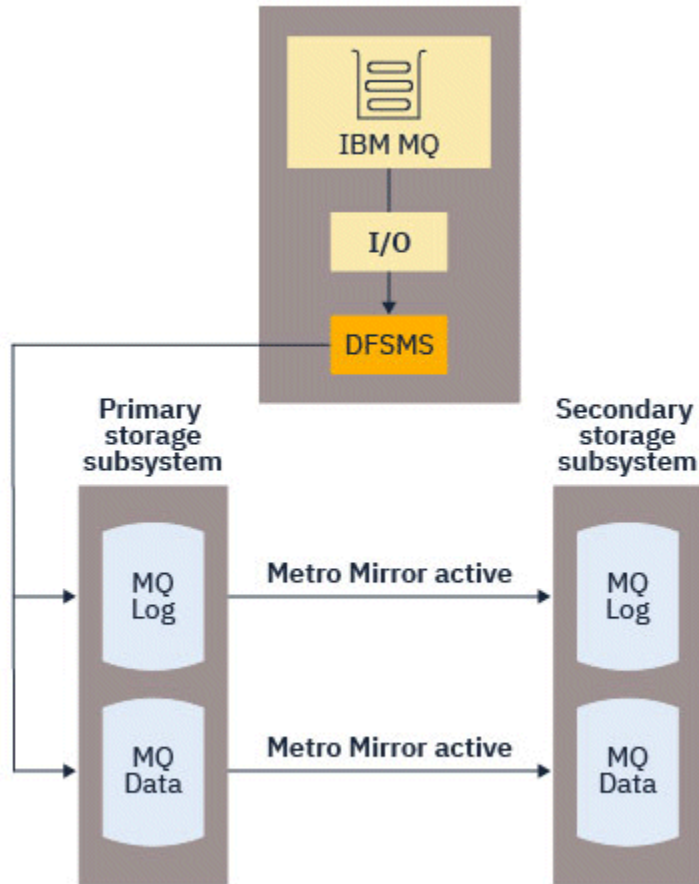
All of the following IBM MQ data set types can be replicated using Metro Mirror. However, exactly which ones are replicated depends on the availability requirements of your enterprise:

- Active logs
- Archive logs
- Bootstrap data set (BSDS)
- Page sets
- Shared message data set (SMDS)
- Data sets used for configuration, for example, in the CSQINP\* DD cards on the MSTR JCL

## Using zHyperWrite with IBM MQ active logs

When a write is made to a data set that is replicated using Metro Mirror, the write is first made to the primary volume, and then replicated to the secondary volume. This replication is done by the storage subsystem and is transparent to the application that issued the write, for example IBM MQ.

This process is illustrated in the following diagram.

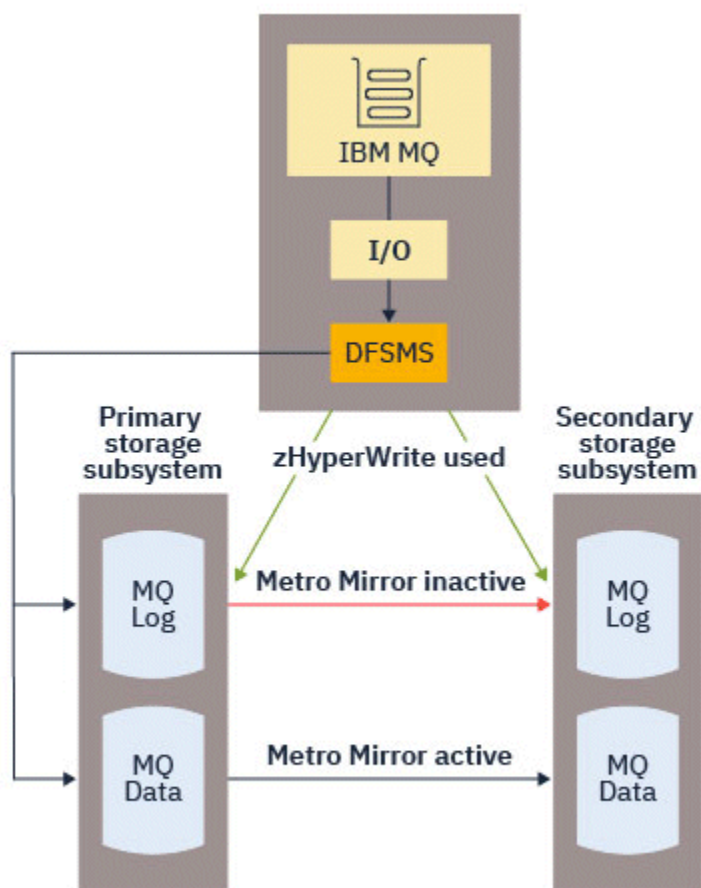


Because both writes to the primary and secondary storage subsystems need to complete before the write returns to IBM MQ, use of Metro Mirror can have a performance impact. You need to balance this performance impact against the availability benefits of using Metro Mirror.

The IBM MQ active logs are most sensitive to the performance impact of using Metro Mirror. IBM MQ allows use of zHyperWrite with the active logs to help reduce this performance impact.

zHyperWrite is a storage subsystem technology that works with z/OS to reduce the performance impact of writes made to data sets that are replicated using Metro Mirror. When zHyperWrite is used, the write to the primary and secondary volumes are issued in parallel at the Data Facility Storage Management Subsystem (DFSMS) level, instead of sequentially at the storage subsystem level, thereby reducing the performance impact.

The following diagram illustrates zHyperWrite being used for the active logs, and Metro Mirror being used for the other IBM MQ data set types. Note that if a zHyperWrite write fails, DFSMS will transparently reissue the write using Metro Mirror.



zHyperWrite on IBM MQ, is supported only on the active log data sets.

In order to use zHyperWrite with the active logs, you need to:

- Configure IBM MQ to use zHyperWrite, and
- The active logs need to be on zHyperWrite capable volumes

You can configure IBM MQ to use zHyperWrite by using one of the following methods:

- Specify `ZHYWRITE(YES)` in the system parameter module.
- Issue the command `SET LOG ZHYWRITE(YES)`.

Set the following conditions for active log data sets to be on zHyperWrite capable volumes:

- Enable the volumes for Metro Mirror, and the volumes support zHyperWrite
- Ensure that the volumes are HyperSwap enabled
- Specify `HYPERWRITE=YES` in the `IECIOSxx` parameter

**> V 9.4.0** Prior to IBM MQ 9.4.0, if all the preceding conditions are met, then writes to the active logs are enabled for zHyperWrite. If one, or more, of these conditions are not met, IBM MQ writes to the active logs as normal, and Metro Mirror replicates the writes if it is configured.

**> V 9.4.0** From IBM MQ 9.4.0, if `ZHYWRITE(YES)` is specified, then IBM MQ always attempts to use zHyperWrite when writing to the active logs, regardless of whether the logs are on zHyperWrite capable volumes. If the logs are not on zHyperWrite capable volumes then Metro Mirror replicates the writes if it is configured. There are no negative effects of attempting to use zHyperWrite if the logs are not on zHyperWrite capable volumes

**Notes:**

- IBM MQ does not require that all active log data sets are on zHyperWrite capable volumes.

If IBM MQ detects that some active log data sets are on zHyperWrite capable volumes, and others are not, it issues message `CSQJ166E` and carries on processing.

- IBM MQ checks whether active log data sets are zHyperWrite capable when the data sets are first opened.

Log data sets are opened either at queue manager start up, or when dynamically adding using the `DEFINE LOG` command. If the log data sets are made zHyperWrite capable while a queue manager has them open, the queue manager will not detect this until it has been restarted.

You can use the output of the `DISPLAY LOG` command to indicate whether the current active log data sets are zHyperWrite capable. The following example shows that both of the data sets are zHyperWrite capable. If the queue manager has been configured with `ZHYWRITE(YES)`, writes to these logs would be enabled for zHyperWrite:

```
Copy %Full zHyperWrite DSName
 1     4 CAPABLE      MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
 2     4 CAPABLE      MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
```

## **Débit des journaux plus rapide avec le lien zHyper**

zHyper La technologie Link est conçue pour réduire la latence d'entrée/sortie (E/S) en fournissant un chemin de communication rapide, fiable et direct entre le processeur et le périphérique d'E/S.

### **Vue générale de zHyper Lien**

zHyperLink peut améliorer le débit des journaux actifs et réduire la durée des transactions IBM MQ jusqu'à 3.5 fois. Cet objectif est atteint en installant zHyper Reliez les adaptateurs sur le z/OS hôte, sélectionnez IBM matériel de stockage et leur connexion à l'aide de zHyper Reliez les câbles. Cela crée une connexion point à point entre le processeur et le périphérique d'E/S, ce qui réduit le temps de réponse des E/S jusqu'à 10 fois par rapport à IBM z FICON® Haute Performance (zHPF). Ce faible temps de réponse est obtenu en utilisant des demandes d'E-S synchrones.

### **Les avantages des E/S synchrones par rapport aux E/S asynchrones**

La tâche de l'enregistreur consiste en une boucle attendant la prochaine donnée qui doit être écrite dans le journal. Lorsque ces données sont disponibles, l'enregistreur planifie l'écriture, attend qu'elle se termine, puis passe à la donnée suivante.

Les E/S traditionnelles sont plus lentes que le processeur. Il est donc plus efficace d'effectuer les E/S de manière asynchrone afin de libérer le processeur pour d'autres tâches. Par conséquent, les E/S asynchrones traditionnelles nécessitent que la tâche de journalisation soit suspendue jusqu'à ce que l'écriture soit terminée. Une fois l'écriture terminée, la tâche de journalisation doit attendre qu'un processeur soit disponible, ce qui ajoute un court délai de redistribution, ainsi que des retards provoqués par le repeuplement du cache du processeur.

zHyper Link fournit des temps d'E/S beaucoup plus rapides, qui sont donc plus proches de la vitesse du processeur, avec zHyper Link, les E/S peuvent être effectuées de manière synchrone, ce qui signifie que la tâche de journalisation n'est pas suspendue pendant l'opération d'écriture, supprimant ainsi les retards liés à la redistribution et au cache.

Pendant l'écriture, la tâche d'enregistrement utilise toujours activement le processeur, ce qui augmente l'utilisation du processeur par rapport aux E/S traditionnelles.

Si le gestionnaire de files d'attente tente d'utiliser zHyper Lien, et l'écriture du lien échoue, par exemple en raison de problèmes de configuration, puis le gestionnaire de files d'attente revient de manière transparente aux E/S traditionnelles.



## Configuration matérielle minimale requise

- IBM z14 ou ultérieure
- DS8880 ou version ultérieure

## Conditions logicielles requises

- zHyperLink Express est pris en charge sur z/OS 2.3 ou version ultérieure.
- L'image z/OS doit s'exécuter dans une partition logique et non en tant qu'invité sous IBM z/VM®.
- zHyperLink requiert l'activation de IBM z High-Performance FICON (zHPF).

## Utilisation de zHyperLink avec les journaux actifs IBM MQ

Pour utiliser zHyper En lien avec les logs actifs d'un gestionnaire de files d'attente, vous devez :

- Configurer IBM MQ utiliser zHyper Lien, et
- Assurez-vous que les journaux actifs sont activés zHyper Lien des volumes compatibles.

Voir [Commencer avec IBM zHyper Lien pour z/OS](#) pour plus d'informations.

Vous pouvez configurer IBM MQ utiliser zHyper Créez un lien en utilisant l'une des méthodes suivantes :

- Spécifiez `ZHYLINK(YES)` dans les paramètres de journal.
- Exécutez la commande `SET LOG ZHYLINK(YES)`.

### Remarques :

- zHyperLink requiert que zHyperWrite soit activé. Cela signifie que pour utiliser ZHYLINK, ZHYWRITE doit également être activé dans les paramètres du journal. Lorsque vous spécifiez uniquement ZHYLINK (YES) lorsque ZHYWRITE (NO) est défini sur le gestionnaire de files d'attente, le paramètre ZHYWRITE est automatiquement remplacé par YES.
- Une tentative explicite de définition de ZHYLINK (YES) avec ZHYWRITE (NO) entraîne l'exécution anormale de la commande SET LOG.
- La définition de ZHYLINK=YES dans les ZPRMs remplace ZHYWRITE par YES.

Si vous rencontrez des problèmes, voir [Traitement des incidents liés à zHyperLink](#) pour plus d'informations.

IBM MQ ne requiert pas que tous les fichiers journaux actifs soient sur des volumes compatibles avec zHyperLink, mais il vous est conseillé de le faire. Si IBM MQ détecte que certains fichiers journaux actifs se trouvent sur des volumes compatibles avec zHyperLink et que d'autres ne le sont pas, il émet le message CSQJ601E et poursuit le traitement.

IBM MQ vérifie si les fichiers journaux actifs sont compatibles avec zHyperLink lorsque les fichiers sont ouverts pour la première fois. Les fichiers journaux sont ouverts au démarrage du gestionnaire de files d'attente ou lors de l'ajout dynamique à l'aide de la commande [DEFINE LOG](#) . Si les fichiers journaux sont rendus compatibles avec zHyperLink alors qu'un gestionnaire de files d'attente les a ouverts, le gestionnaire de files d'attente ne les détecte pas tant qu'il n'a pas été redémarré.

Si ZHYLINK (YES) est spécifié, IBM MQ tente toujours d'utiliser le lien zHyper lors de l'écriture dans les journaux actifs, que les journaux soient ou non sur des volumes compatibles avec zHyperLink. Il n'y a aucun effet négatif de la tentative d'utilisation de zHyperLink si les journaux ne sont pas sur des volumes compatibles avec zHyperLink.

Vous pouvez utiliser la sortie de la commande [DISPLAY LOG](#) pour indiquer le statut du lien zHyper pour les fichiers journaux actifs en cours:

```
Copy %Full zHyperWrite Encrypted DSName
 1 81 YES NO MQTST.SUBSYS.MQDL.LOGCOPY1.DS001
 2 81 YES NO MQTST.SUBSYS.MQDL.LOGCOPY2.DS001
Copy zHyperLink
 1 YES
```

Le statut de la liaison zHyperest l'un des suivants:

**YES**

La liaison zHyperest activée sur le gestionnaire de files d'attente et sera tentée sur toutes les écritures.

**NO**

zHyperLink n'est pas activé sur le gestionnaire de files d'attente et le fichier **n'est pas** sur les volumes compatibles avec zHyperLink.

**capable**

La liaison zHyper n'est pas activée sur le gestionnaire de files d'attente et le fichier **est** sur un volume compatible avec la liaison zHyper.

Il existe plusieurs statistiques SMF supplémentaires pour la surveillance et la compréhension zHyper Performances des liens ; voir [zHyper Statistiques des liens](#) pour plus de détails.

## Séances d'écriture

Lors de l'utilisation zHyper Lien, un ou plusieurs zHyper Des sessions d'écriture de lien sont établies avec le DASD. Le DASD actuel prend en charge un maximum de 64 sessions d'écriture simultanées. Vous devez donc soigneusement réfléchir aux gestionnaires de files d'attente que vous activez. zHyper Lien activé et si d'autres sous-systèmes, tels que Db2 utilisent également zHyper Lien pour écrire sur le même DASD. Si vous manquez de sessions d'écriture disponibles, le gestionnaire de files d'attente revient automatiquement à l'utilisation des E/S asynchrones traditionnelles.

Vous pouvez calculer le nombre de zHyper Liez les sessions d'écriture comme suit :

Number of log copies (either 1 or 2) \* number of stripes per log copy \* 2  
if Metro Mirror (PPRC) is used.

Par conséquent, un gestionnaire de files d'attente en mode de journalisation unique avec une seule bande et aucune Metro Mirror utilise une seule session d'écriture. Un gestionnaire de files d'attente en mode double journalisation, avec deux bandes et PPRC utilise 8 sessions d'écriture.

**Remarque :** Alors que Metro Mirror entraîne l'utilisation de deux fois plus de sessions d'écriture, ces sessions d'écriture sont réparties également entre les deux DASD en miroir.

## **Planning your log archive storage**

Use this topic to understand the different ways of maintaining your archive log data sets.

You can place archive log data sets on standard-label tapes, or DASD, and you can manage them by data facility hierarchical storage manager (DFHSM). Each z/OS logical record in an archive log data set is a VSAM control interval from the active log data set. The block size is a multiple of 4 KB.

Archive log data sets are dynamically allocated, with names chosen by IBM MQ. The data set name prefix, block size, unit name, and DASD sizes needed for such allocations are specified in the system parameter module. You can also choose, at installation time, to have IBM MQ add a date and time to the archive log data set name.

It is not possible to specify with IBM MQ, specific volumes for new archive logs, but you can use Storage Management routines to manage this. If allocation errors occur, offloading is postponed until the next time offloading is triggered.

If you specify dual archive logs at installation time, each log control interval retrieved from the active log is written to two archive log data sets. The log records that are contained in the pair of archive log data sets are identical, but the end-of-volume points are not synchronized for multivolume data sets.

## **Should your archive logs reside on tape or DASD?**

When deciding whether to use tape or DASD for your archive logs, there are a number of factors that you should consider:

- Review your operating procedures before deciding about tape or disk. For example, if you choose to archive to tape, there must be enough tape drive when they are required. After a disaster, all subsystems might want tape drives and you might not have as many free tape drives as you expect.
- During recovery, archive logs on tape are available as soon as the tape is mounted. If DASD archives have been used, and the data sets migrated to tape using hierarchical storage manager (HSM), there is a delay while HSM recalls each data set to disk. You can recall the data sets before the archive log is used. However, it is not always possible to predict the correct order in which they are required.
- When using archive logs on DASD, if many logs are required (which might be the case when recovering a page set after restoring from a backup) you might require a significant quantity of DASD to hold all the archive logs.
- In a low-usage system or test system, it might be more convenient to have archive logs on DASD to eliminate the need for tape mounts.
- Both issuing a RECOVER CFSTRUCT command, and backing out a persistent unit of work, result in the log being read backwards. Tape drives with hardware compression perform badly on operations that read backwards. Plan sufficient log data on DASD to avoid reading backwards from tape.

Archiving to DASD offers faster recoverability but is more expensive than archiving to tape. If you use dual logging, you can specify that the primary copy of the archive log go to DASD and the secondary copy go to tape. This increases recovery speed without using as much DASD, and you can use the tape as a backup.

See “Changing the storage medium for archive logs” on page 180 for details of how you archive your logs from tape to DASD, and how you carry out the reverse process.

### **Archiving to tape**

If you choose to archive to a tape device, IBM MQ can extend to a maximum of 20 volumes.

If you are considering changing the size of the active log data set so that the set fits on one tape volume, note that a copy of the BSDS is placed on the same tape volume as the copy of the active log data set. Adjust the size of the active log data set downward to offset the space required for the BSDS on the tape volume.

If you use dual archive logs on tape, it is typical for one copy to be held locally, and the other copy to be held off-site for use in disaster recovery.

### **Archiving to DASD volumes**

IBM MQ requires that you catalog all archive log data sets allocated on non-tape devices (DASD). If you choose to archive to DASD, the CATALOG parameter of the CSQ6ARVP macro must be YES. If this parameter is NO, and you decide to place archive log data sets on DASD, you receive message CSQJ072E each time an archive log data set is allocated, although IBM MQ still catalogs the data set.

If the archive log data set is held on DASD, the archive log data sets can extend to another volume; multivolume is supported.

If you choose to use DASD, make sure that the primary space allocation (both quantity and block size) is large enough to contain either the data coming from the active log data set, or that from the corresponding BSDS, whichever is the larger of the two.

This minimizes the possibility of unwanted z/OS X' B37 ' or X' E37 ' abend codes during the offload process. The primary space allocation is set with the PRIQTY (primary quantity) parameter of the CSQ6ARVP macro.

Archive log data sets can exist on large or extended-format sequential data sets. SMS ACS routines now use DSNTYPE(LARGE) or DSNTYPE(EXT).

IBM MQ supports allocation of archive logs as extended format data sets. When extended format is used, the maximum archive log size is increased from 65535 tracks to the maximum active log size

of 4GB. Archive logs are eligible for allocation in the extended addressing space (EAS) of extended address volumes (EAV).

Where the required hardware and software levels are available, allocating archive logs to a data class defined with COMPACTION using zEDC might reduce the disk storage required to hold archive logs. For more information, see [IBM MQ for z/OS: Reducing storage occupancy with IBM zEnterprise Data Compression \(zEDC\)](#) and [zEnterprise Data Compression \(zEDC\)](#) for more information.

The z/OS data set encryption feature can be applied to archive logs for queue managers running on IBM MQ. These archive logs must be allocated through Automatic Class Selection (ACS) routines to a data class defined with EXTENDED attributes, and a data set key label that ensures the data is AES encrypted.

## Using SMS with archive log data sets

If you have MVS/DFP storage management subsystem ( DFSMS) installed, you can write an Automatic Class Selection (ACS) user-exit filter for your archive log data sets, which helps you convert them for the SMS environment.

Such a filter, for example, can route your output to a DASD data set, which DFSMS can manage. You must exercise caution if you use an ACS filter in this manner. Because SMS requires DASD data sets to be cataloged, you must make sure the CATALOG DATA field of the [CSQ6ARVP](#) macro contains YES. If it does not, message [CSQJ072E](#) is returned; however, the data set is still cataloged by IBM MQ.

For more information about ACS filters, see [Data sets that DFSMSHsm dynamically allocates during aggregate backup processing](#).

## Changing the storage medium for archive logs

The procedure for changing the storage medium used by archive logs.

## About this task

This task describes how to change the storage medium used for archive logs, for example moving from archiving to tape to archiving to DASD.

You have a choice of how to make the changes:

1. Make the changes only using the [CSQ6ARVP](#) macro so that they are applied from the next time the queue manager restarts.
2. Make the changes using the [CSQ6ARVP](#) macro, and dynamically using the [SET ARCHIVE](#) command. This means that the changes apply from the next time the queue manager archives a log file, and persist after the queue manager restarts.

## Procedure

1. Changing so archive logs are stored on DASD instead of tape:
  - a) Read the section [“Archiving to DASD volumes”](#) on page 179 and review the [CSQ6ARVP](#) parameters.
  - b) Make changes to the following parameters in [CSQ6ARVP](#)
    - Update the UNIT and, if necessary, the UNIT2 parameters.
    - Update the BLKSIZE parameter, as the optimal setting for DASD differs from tape.
    - Set the PRIQTY and SECQTY parameters to be large enough to hold the largest of the active log or BSDS.
    - Set the CATALOG parameter to be YES.
    - Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
    - Set the ARCWTOR parameter to NO if it is not already.
2. Changing so archive logs are stored on tape instead of DASD:

- a) Read the section “Archiving to tape” on page 179, and review the CSQ6ARVP parameters.
- b) Make changes to the following parameters in CSQ6ARVP:
  - Update the UNIT and, if necessary, the UNIT2 parameters.
  - Update the BLKSIZE parameter, as the optimal setting for tape differs from DASD.
  - Confirm the ALCUNIT setting is what you want. You should use BLK, because it is independent of the device type.
  - Review the setting of the ARCWTOR parameter.

## **How long do I need to keep archive logs**

Use the information in this section to help you plan your backup strategy.

You specify how long archive logs are kept in days, using the ARCRETN parameter in [USING CSQ6ARVP](#) or the [SET SYSTEM](#) command. After this period the data sets can be deleted by z/OS.

You can manually delete archive log data sets when they are no longer needed.

- The queue manager might need the archive logs for recovery.
 

The queue manager can only keep the most recent 1000 archives in the BSDS. When the archive logs are not in the BSDS they cannot be used for recovery, and are only of use for audit, analysis, or replay type purposes.
- You might want to keep the archive logs so that you can extract information from the logs. For example, extracting messages from the log, and reviewing which user ID put or got the message.

The BSDS contains information on logs and other recovery information. This data set is a fixed size. When the number of archive logs reaches the value of [MAXARCH](#) in CSQ6LOGP, or when the BSDS fills up, the oldest archive log information is overwritten.

There are utilities to remove archive log entries from the BSDS, but in general, the BSDS wraps and overlays the oldest archive log record.

### **When is an archive log needed**

You need to back up your page sets regularly. The frequency of backups determines which archive logs are needed in the event of losing a page set.

You need to back up your CF structures regularly. The frequency of backups determines which archive logs are needed in the event of losing data in the CF structure.

The archive log might be needed for recovery. The following information explains when the archive log might be needed, where there are problems with different IBM MQ resources.

#### **Loss of a page set**

You must recover your system from your backup and restart the queue manager.

You need the logs from when the backup was taken, as well as up to three log data sets prior to the backup being taken.

#### **All LPARs lose connectivity to a CF structure, or the structure is unavailable**

Use the [RECOVER CFSTRUCT](#) command to recover the structure.

Structure recovery requires the logs from all queue managers that have accessed the structure since the last backup (back to the time when the backup was taken) plus the structure backup itself in the log of the queue manager that took the backup.

If you have been doing frequent backups of the CF structures, the data should be in active logs, and you should not need archive logs.

If there is no recent backup of the CF structure, you might need archive logs.

**Note:** All non persistent messages will be lost; all persistent messages will be re-created by performing the following tasks:

1. Reading the last CF structure backup from the log
2. Reading the logs from all queue managers that have used the structure
3. Merging updates since the backup

### Administration structure rebuild

If you need to rebuild the administration structure, the information is read from the last checkpoint of the log for each queue manager in the QSG.

If a queue manager is not active, another queue manager in the QSG reads the log.

You should not need archive logs.

### Loss of an SMDS data set

If you lose an SMDS data set, or the data set gets corrupted, the data set becomes unusable and the status for it is set to FAILED. The CF structure is unchanged.

In order to restore the SMDS data set, you need to:

1. Redefine the SMDS data set, and
2. Recover the CF structure by issuing the [RECOVER CFSTRUCT](#) command.

**Note:** All non persistent messages on the CF structure will be lost; all persistent messages will be restored.

The requirement for queue manager logs is the same as for recovering from a structure that is unavailable.

## **Planning to increase the maximum addressable log range**

You can increase the maximum addressable log range by configuring your queue manager to use a larger log relative byte address (RBA).

The log RBA size was increased from IBM MQ for z/OS 8.0. For an overview of this change, see [Larger log Relative Byte Address](#).

Queue managers created at IBM MQ 9.3.0 or later, have 8 byte log RBA enabled by default and, therefore, do not require conversion.

You can convert your queue managers to use 8 byte log RBA values at any time. A queue sharing group can contain some queue managers with 8 byte log RBA enabled, and some queue managers with 6 byte log RBA enabled.

### Undoing the change

The change cannot be backed out.

### How long does it take?

The change requires a queue manager restart. Stop the queue manager, run the CSQJUCNV utility against the bootstrap data set (BSDS), or data sets, to create new data sets, rename these bootstrap data sets, and restart the queue manager. The CSQJUCNV utility usually takes a few seconds to run.

### What impact does this have?

- With 8 byte log RBA in use, every write of data to the log data sets has additional bytes. Therefore, for a workload consisting of persistent messages there is a small increase in the amount of data written to the logs.
- Data written to a page set, or coupling facility (CF) structure, is not affected.

### Related tasks

[Implementing the larger log Relative Byte Address](#)

## Planning your channel initiator

The channel initiator provides communications between queue managers, and runs in its own address space.

There are two types of connections:

1. Application connections to a queue manager over a network. These are known as client channels.
2. Queue manager to queue manager connections. These are known as MCA channels.

### Listeners

A channel listener program listens for incoming network requests and starts the appropriate channel when that channel is needed. To process inbound connections the channel initiator needs at least one IBM MQ listener task configured. A listener can either be a TCP listener, or a LU 6.2 listener.

Each listener requires a TCP port or LU name.

Note that you can have more than one listener for each channel initiator.

### TCP/IP

A channel initiator can operate with more than one TCP stack on the same z/OS image. For example, one TCP stack could be for internal connections, and another TCP stack for external connections.

When you define an output channel:

1. You set the destination host and port of the connection. This can be either:
  - an IP address, for example 10.20.4.6
  - a host name, for example mvs-prod.myorg.com

If you use a host name to specify the destination, IBM MQ uses the Domain Name System (DNS) to resolve the IP address of the destination.

2. If you are using multiple TCP stacks you can specify the **LOCLADDR** parameter on the channel definition, which specifies the IP Stack address to be used.

You should plan to have a highly available DNS server, or DNS servers. If the DNS is not available, outbound channels might not be able to start, and channel authentication rules that map an incoming connection using a host name cannot be processed.

### APPC and LU 6.2

If you are using APPC, the channel initiator needs an LU name, and configuration in APPC.

### Queue sharing groups

To provide a single system image, and allow an incoming IBM MQ connection request to go to any queue manager in the queue sharing group, you need to do some configuration. For example:

1. A hardware network router. This router has one IP address seen by the enterprise, and can route the initial request to any queue manager connected to this hardware.
2. A Virtual IP address (VIP). An enterprise wide IP address is specified, and that address can be routed to any one of the TCP stacks in a sysplex. The TCP stack can then route it to any listening queue manager in the sysplex.

### Protecting IBM MQ traffic

You can configure IBM MQ to use TLS connections to protect data on the wire. To use TLS you need to use digital certificates and key rings.

You also need to work with the personnel at the remote end of the channel, to ensure that you have compatible IBM MQ definitions and compatible certificates.

You can control which connections can connect to IBM MQ and the user ID, based on

- IP address
- Client user ID
- Remote queue manager, or
- Digital certificate (see [Channel Authentication Records](#))

It is also possible to restrict client applications by ensuring that they supply a valid user ID and password (see [Connection Authentication](#)).

You can get the channel initiator working, and then configure each channel to use TLS, one at a time.

## Monitoring the channel initiator

There are MQSC commands that give information about the channel initiator and channels:

- The [DISPLAY CHINIT](#) command gives information about the channel initiator, and active listeners.
- The [DISPLAY CHSTATUS](#) command displays the activity and status of a channel.

The channel initiator can also produce SMF records with information about the channel initiator tasks and channel activity. See [“Planning for channel initiator SMF data” on page 185](#) for more information.

The channel initiator emits messages to the job log when channels start and stop. Automation in your enterprise can use these messages to capture status. As some channels are active for only a few seconds, many messages can be produced. You can suppress these messages either by using the z/OS message processing facility, or by setting **EXCLMSG** with the [SET SYSTEM](#) command.

## Configuring your IBM MQ channel definitions

When you have many queue managers connected together it can be hard to manage all the object definitions. Using IBM MQ clustering can simplify this.

You specify two queue managers as full repositories. Other queue managers need one connection to, and one connection from, one of the repositories. When connections to other queue managers are needed, the queue manager creates and starts channels automatically.

If you are planning to have a large number of queue managers in a cluster, you should plan to have queue managers that act as dedicated repositories and have no application traffic.

See [“Planification de vos files d'attente réparties et de vos clusters” on page 21](#) for more information.

## Actions before you configure the channel initiator

1. Decide if you are using TCP/IP or APPC.
2. If you are using TCP, allocate at least one port for IBM MQ.
3. If you need a a DNS server, configure the server to be highly available if required.
4. If you are using APPC, allocate an LU name, and configure APPC.

## Actions after you have configured the channel initiator, before you go into production

1. Plan what connections you will have:
  - a. Client connections from remote applications.
  - b. MCA channels to and from other queue managers. Typically you have a channel to and from each remote queue manager.
2. Set up clustering, or join an existing clustering environment.



3. Consider whether you need to use multiple TCP stacks, VIPA, or an external router for availability in front of the channel initiator.
4. If you are planning on using TLS:
  - a. Set up the key ring
  - b. Set up certificates
5. If you are planning on using channel authentication:
  - a. Decide the criteria for mapping inbound sessions to MCA user IDs
  - b. Enable reverse DNS lookup by setting the queue manager parameter **REVDNS**
  - c. Review security. For example, delete the default channels, and specify user IDs with only the necessary authority in the **MCAUSER** attribute for a channel.
6. Capture the accounting and statistics SMF records produced by the channel initiator and post process them.
7. Automate the monitoring of job log messages.
8. If necessary, tune your network environment to improve throughput. With TCP, large send and receive buffers improve throughput. You can force MQ to use specific TCP buffer sizes using the commands:

```
RECOVER QMGR(TUNE CHINTCPBDYNSZ nnnnn)
RECOVER QMGR(TUNE CHINTCPSBDYNSZ nnnnn)
```

which sets the SO\_RCVBUF, and SO\_SNDBUF, for the channels to the size in bytes specified in nnnnn.

### Related concepts

[“Planning for your queue manager” on page 154](#)

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

## Planning for channel initiator SMF data

You need to plan the implementation of collecting SMF data for the channel initiator.

The channel initiator produces two types of record:

- Statistics data with information about the channel initiator and the tasks within it.
- Channel accounting data with information similar to the [DISPLAY CHSTATUS](#) command.

You start collecting statistics data using the command:

```
START TRACE(STAT) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(STAT) CLASS(4)
```

You start collecting accounting data using the command:

```
START TRACE(ACCTG) CLASS(4)
```

and stop it using the command:

```
STOP TRACE(ACCTG) CLASS(4)
```

You can control which channels have accounting data collected for using the **STATCHL** attribute on the channel definition or the queue manager.

- For client channels, you must set **STATCHL** at the queue manager level.

- For automatically defined cluster sender channels, you can control the collection of accounting data with the **STATACLS** queue manager attribute.

The default value of **STATCHL** for the queue manager is OFF. In order to collect channel accounting data you must change the value of **STATCHL** from the default on either the queue manager or channel definition, in addition to starting class 4 accounting trace.

The SMF records are produced when:

- From IBM MQ for z/OS 9.3.0 onwards, the time interval indicated by the CSQ6SYSP **STATIME** or **ACCTIME** parameters has elapsed; or, if **STATIME** or **ACCTIME** is zero on the SMF data collection broadcast. The requests to collect SMF data for the channel initiator and the queue manager are synchronized.
- A STOP TRACE(ACCTG) CLASS(4) or STOP TRACE(STAT) CLASS(4) command is issued, or
- The channel initiator is shut down. At this point any SMF data is written out.

If a channel stops during the SMF interval, accounting data is written to SMF the next time the SMF processing runs. If a client connects, does some work and disconnects, then reconnects and disconnects, there are two sets of channel accounting data produced.

The statistics data normally fits into one SMF record, however, multiple SMF records might be created if a large number of tasks are in use.

Accounting data is gathered for each channel for which it is enabled, and normally fits into one SMF record. However, multiple SMF records might be created if a large number of channels are active.

The cost of collecting the channel initiator SMF data is small. Typically the increase in CPU usage is under a few percent, and often within measurement error.

Before you use this function you need to work with your z/OS systems programmer to ensure that SMF has the capacity for the additional records, and that they change their processes for extracting SMF records to include the new SMF data.

For channel initiator statistics data, the SMF record type is 115 and sub-type 231.

For channel initiator accounting data, the SMF record type is 116 and sub-type 10.

You can write your own programs to process this data, or use the SupportPac [MP1B](#) that contains a program, MQSMF, for printing the data, and creating data in Comma Separated Values (CSV) format suitable for importing into a spread sheet.

If you are experiencing issues with capturing channel initiator SMF data, see [Dealing with issues when capturing SMF data for the channel initiator \(CHINIT\)](#) for further information.

### Related tasks

[Interpreting IBM MQ performance statistics](#)

[Troubleshooting channel accounting data](#)

## Planning your z/OS TCP/IP environment

To get the best throughput through your network, you must use TCP/IP send and receive buffers with a size of 64 KB, or greater. With this size, the system optimizes its buffer sizes.

See [What is Dynamic Right Sizing for High Latency Networks?](#) for more information.

You can check your system buffer size by using the following Netstat command, for example:

```
TSO NETSTAT ALL (CLIENT csq1CHIN
```

The results display much information, including the following two values:

```
ReceiveBufferSize: 0000065536
SendBufferSize: 0000065536
```

65536 is 64 KB. If your buffer sizes are less than 65536, you must work with your network team to increase the **TCPSENDBFRSIZE** and **TCPRCVBUFRSIZE** values in the PROFILE DDName in the TCPIP procedure. For example, you might use the following command:

```
TCPCONFIG TCPSENDBFRSIZE 65536 TCPRCVBUFRSIZE 65536
```

If you are unable to change your system-wide **TCPSENDBFRSIZE** or **TCPRCVBUFRSIZE** settings, contact your IBM Software Support center.

## **Planning your queue sharing group (QSG)**

The easiest way to implement a shared queuing environment, is to configure a queue manager, add that queue manager to a QSG, then add other queue managers to the QSG.

A queue sharing group uses Db2 tables to store configuration information. There is one set of tables used by all QSGs that share the same Db2 data sharing group.

Shared queue messages are stored in a structure in a coupling facility (CF). Each QSG has its own set of CF structures. You need to configure the structures to meet your needs.

Messages over 63KB in size cannot be stored in the CF. You need to use either Shared Message Data Sets (SMDS) or Db2 for these messages.

### **Message profiles and capacity planning**

You should understand the message profile of your shared queue messages. The following are examples of factors that you need to consider:

- Average, and maximum message size
- The typical queue depth, and exception queue depth. For example, you might need to have enough capacity to hold messages for a whole day, and the typical queue depth is under 100 messages.

If the message profile changes, you can increase the size of the structures, or implement SMDS, at a later date.

If you want to be able to handle a large peak volume of messages, you can configure IBM MQ to offload messages to SMDS when the usage of the structure reaches user specified thresholds.

You need to decide if you want to duplex the CF structures. This is controlled by the CF structure definition in the CFRM policy:

1. A duplexed structure uses two coupling facilities. If there is a problem with one CF, there is no interruption to the service, and the structure can be rebuilt on a third CF, if one is available. Duplexed structures can significantly impact the performance of operations on shared queues.
2. If the structure is not duplexed, then a problem with the CF means that shared queues on structures in that CF will become unavailable until the structure can be rebuilt in another CF.

IBM MQ can be configured to automatically rebuild structures in another CF in this case. Persistent messages will be recovered from the logs of the queue managers.

Note that it is easy to change the CF definitions.

You can define a structure so that it can hold nonpersistent messages only, or so that it can hold persistent and nonpersistent messages.

Structures that can hold persistent messages need to be backed up periodically. Back up your CF structures at least every hour to minimize the time needed to recover the structure in the event of a failure. The backup is stored in the log data set of the queue manager performing the backup.

If you are expecting to have a high throughput of messages on your shared queues, it is best practice to have a dedicated queue manager for backing up the CF structures. This reduces the time needed to recover the structures, as a less data needs to be read from queue manager logs.

## Channels

To provide a single system image for applications connecting into an IBM MQ QSG, you can define shared input channels. If these are set up, then a connection coming into the queue sharing group environment, can go to any queue manager in the QSG.

You might need to set up a network router, or Virtual IP address (VIPA) for these channels.

You can define shared output channels. A shared output channel instance can be started from any queue manager in the QSG.

See [Shared channels](#) for more information.

## Security

You protect IBM MQ resources using an external security manager. If you are using RACF®, the RACF profiles are prefixed with the queue manager name. For example, a queue named APPLICATION.INPUT would be protected using a profile in the MQQUEUE class named qmqzName . APPLICATION . INPUT .

When using a queue sharing group you can continue to protect resources with profiles prefixed with the queue manager name, or you can prefix profiles with the queue sharing group name. For example qsgName . APPLICATION . INPUT .

You should aim to use profiles prefix with the queue sharing group name because this means there is a single definition for all queue managers, saving you work, and preventing a mismatch in definitions between queue managers.

## Related concepts

[“Planning for your queue manager” on page 154](#)

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

## **Planning your coupling facility and offload storage environment**

Use this topic when planning the initial sizes, and formats of your coupling facility (CF) structures, and shared message data set (SMDS) environment or Db2 environment.

This section contains information about the following topics:

- [“Defining coupling facility resources” on page 188](#)
  - [Deciding your offload storage mechanism](#)
  - [Planning your structures](#)
  - [Planning the size of your structures](#)
  - [Mapping shared queues to structures](#)
- [“Planning your shared message data set \(SMDS\) environment” on page 194](#)
- [“Planning your Db2 environment” on page 197](#)

## Defining coupling facility resources

If you intend to use shared queues, you must define the coupling facility structures that IBM MQ will use in your CFRM policy. To do this you must first update your CFRM policy with information about the structures, and then activate the policy.

Your installation probably has an existing CFRM policy that describes the coupling facilities available. The Administrative data utility is used to modify the contents of the policy based on textual statements you provide. You must add statements to the policy that defines the names of the new structures, the coupling facilities that they are defined in, and what size the structures are.

The CFRM policy also determines whether IBM MQ structures are duplexed and how they are reallocated in failure scenarios. [Shared queue recovery](#) contains recommendations for configuring CFRM for resilience to failures that affect the coupling facility.

## Deciding your offload storage environment

The message data for shared queues can be offloaded from the coupling facility and stored in either a Db2 table or in an IBM MQ managed data set called a *shared message data set* (SMDS). Messages which are too large to store in the coupling facility (that is, larger than 63 KB) must always be offloaded, and smaller messages can optionally be offloaded to reduce coupling facility space usage.

For more information, see [Specifying offload options for shared messages](#).

## Planning your structures

A queue sharing group (QSG) requires a minimum of two structures to be defined. The first structure, known as the administrative structure, is used to coordinate IBM MQ internal activity across the queue sharing group. No user data is held in this structure. It has a fixed name of *qsg-nameCSQ\_ADMIN* (where *qsg-name* is the name of your queue sharing group). Subsequent structures are known as application structures, and are used to hold the messages on IBM MQ shared queues. Each structure can hold up to 512 shared queues.

An application structure named *qsg-nameCSQSYSAPPL* is used for system queues. Defining this structure is optional, but it is required in order to use certain features. By default, the `SYSTEM.QSG.CHANNEL.SYNCQ` and `SYSTEM.QSG.UR.RESOLUTION.QUEUE` queues are defined on the *qsg-nameCSQSYSAPPL* structure.

### Using multiple structures

A queue sharing group can connect to up to 64 coupling facility structures. One of these structures must be the administration structure. If it is defined, another of these structures might be the *qsg-nameCSQSYSAPPL* structure. You can use up to 63 (62 if *qsg-nameCSQSYSAPPL* is defined) structures for message data. You might choose to use multiple application structures for any of the following reasons:

- You have some queues that are likely to hold a large number of messages and so require all the resources of an entire coupling facility.
- You have a requirement for a large number of shared queues, so they must be split across multiple structures because each structure can contain only 512 queues.
- RMF reports on the usage characteristic of a structure suggest that you should distribute the queues it contains across a number of coupling facilities.
- You want some queue data to be held in a physically different coupling facility from other queue data for data isolation reasons.
- Recovery of persistent shared messages is performed using structure level attributes and commands, for example `BACKUP CFSTRUCT`. To simplify backup and recovery, you could assign queues that hold nonpersistent messages to different structures from those structures that hold persistent messages.

When choosing which coupling facilities to allocate the structures in, consider the following points:

- Your data isolation requirements.
- The volatility of the coupling facility (that is, its ability to preserve data through a power outage).
- Failure independence between the accessing MQ systems and the coupling facility, or between coupling facilities.
- The level of coupling facility control code (CFCC) installed on the coupling facility ( IBM MQ requires Level 9 or higher).

## Planning the size of your structures

### The administrative structure

The administrative structure (*qsg-name*CSQ\_ADMIN) must be large enough to contain 1000 list entries for each queue manager in the queue sharing group. When a queue manager starts, the structure is checked to see if it is large enough for the number of queue managers currently *defined* to the queue sharing group. Queue managers are considered as being defined to the queue sharing group if they have been added by the CSQ5PQSG utility. You can check which queue managers are defined to the group with the MQSC `DISPLAY GROUP` command.

**Note:** When calculating the size of the structure, you should allow for the size of large units of work, in addition to the number of queue managers in the queue sharing group.

Table 22 on page 190 shows the minimum required size for the administrative structure for various numbers of queue managers defined in the queue sharing group. These sizes were established for a CFCC level 14 coupling facility structure; for higher levels of CFCC, they probably need to be larger.

Number of queue managers defined in queue sharing group	Required storage
1	6144 KB
2	6912 KB
3	7976 KB
4	8704 KB
5	9728 KB
6	10496 KB
7	11520 KB
8	12288 KB
9	13056 KB
10	14080 KB
11	14848 KB
12	15616 KB
13	16640 KB
14	17408 KB
15	18176 KB
16	19200 KB
17	19968 KB
18	20736 KB
19	21760 KB
20	22528 KB
21	23296 KB
22	24320 KB
23	25088 KB

<i>Table 22. Minimum administrative structure sizes (continued)</i>	
<b>Number of queue managers defined in queue sharing group</b>	<b>Required storage</b>
24	25856 KB
25	27136 KB
26	27904 KB
27	28672 KB
28	29696 KB
29	30464 KB
30	31232 KB
31	32256 KB

When you add a queue manager to an existing queue sharing group, the storage requirement might have increased beyond the size recommended in [Table 22 on page 190](#). If so, use the following procedure to estimate the required storage for the *qsg-name*CSQ\_ADMIN structure:

1. Issue MQSC command **DISPLAY CFSTATUS(CSQ\_ADMIN)** on an existing member of the queue sharing group.
2. Extract the ENTSMAX information for the CSQ\_ADMIN structure.
3. If this number is less than 1000 times the total number of queue managers you want to define in the queue sharing group, increase the structure size.

### **Application structures**

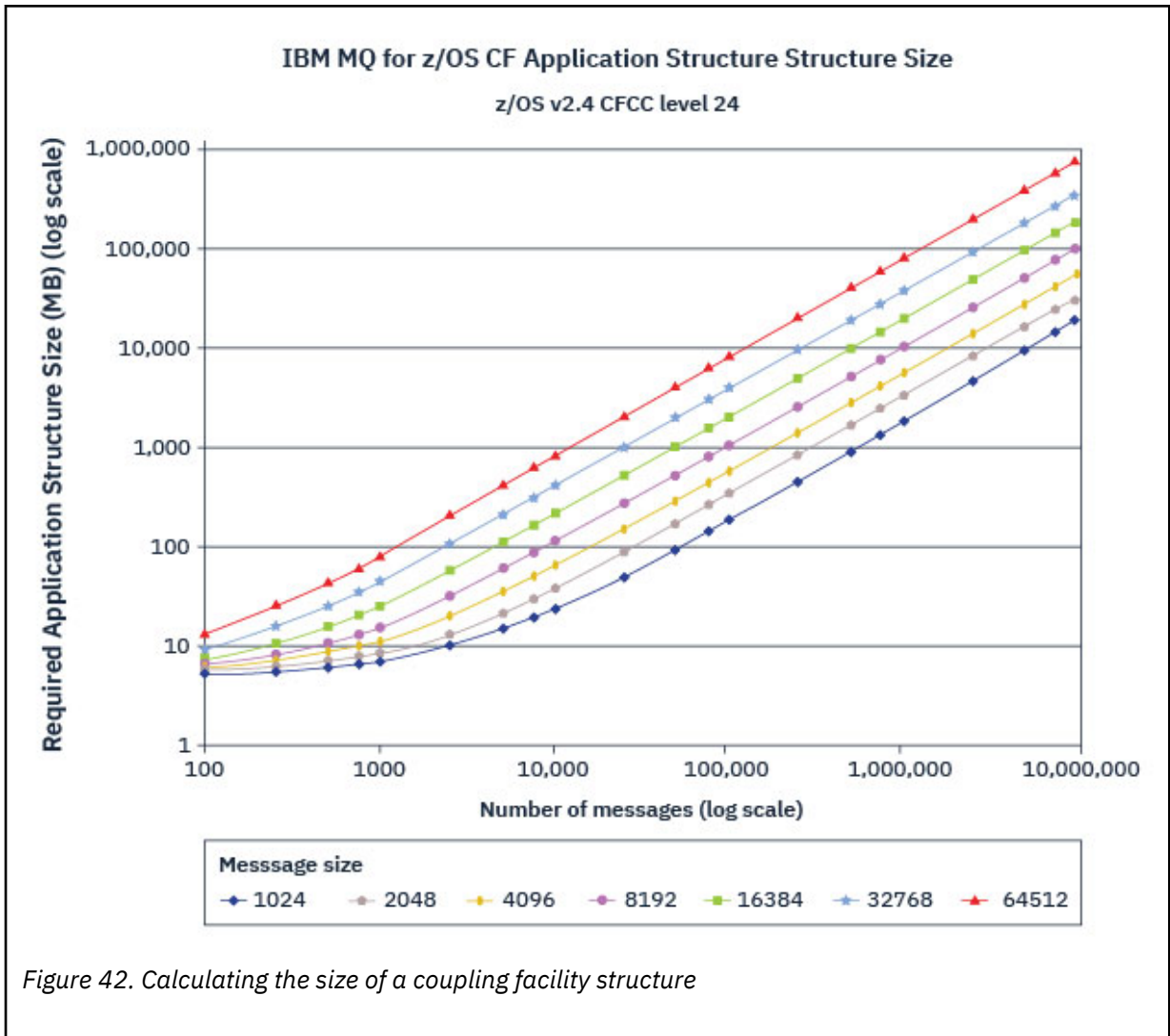
The size of the application structures required to hold IBM MQ messages depends on the likely number and size of the messages to be held on a structure concurrently.

The graph in [Figure 42 on page 192](#) shows how large you should make your CF structures to hold the messages on your shared queues. To calculate the allocation size you need the following information:

- The average size of messages on your queues.
- The total number of messages likely to be stored in the structure.

Find the number of messages along the horizontal axis. Select the curve that corresponds to your message size and determine the required value from the vertical axis. For example, for 200 000 messages of length 1 KB gives a value in the range 256 through 512 MB.

[Table 23 on page 192](#) provides the same information in tabular form.



Use this table to help calculate how large to make your coupling facility structures:

*Table 23. Calculating the size of a coupling facility structure*

Number of messages	1 KB	2 KB	4 KB	8 KB	16 KB	32 KB	63 KB
100	6 MB	6 MB	7 MB	7 MB	8 MB	10 MB	14 MB
1000	8 MB	9 MB	12 MB	17 MB	27 MB	48 MB	88 MB
10000	25 MB	38 MB	64 MB	115 MB	218 MB	423 MB	821 MB
100000	199 MB	327 MB	584 MB	1097 MB	2124 MB	4177 MB	8156 MB

Your CFRM policy should include the following statements:

- INITSIZE is the size in KB that the structure is allocated with when the first queue manager connects to it.
- SIZE is the maximum size that the structure can attain.
- FULLTHRESHOLD sets the percentage value of the threshold at which z/OS issues message IXC585E to indicate that the structure is getting full.



A best practice is to ensure that INITSIZE and SIZE are within a factor of 2. For example, with the figures determined previously, you might include the following statements:

```
STRUCTURE NAME(structure-name)
INITSIZE(value from graph in KB, that is, multiplied by 1024)
SIZE(something larger)
FULLTHRESHOLD(85)
```

```
STRUCTURE NAME(QSG1APPLICATION1)
INITSIZE(262144) /* 256 MB */
SIZE(524288) /* 512 MB */
FULLTHRESHOLD(85)
```

If the structure use reaches the threshold where warning messages are issued, intervention is required. You might use IBM MQ to inhibit MQPUT operations to some of the queues in the structure to prevent applications from writing more messages, start more applications to get messages from the queues, or quiesce some of the applications that are putting messages to the queue.

Alternatively, you can use z/OS facilities to alter the structure size in place. The following z/OS command:

```
SETXCF START,ALTER,STRNAME=structure-name,SIZE=newsize
```

alters the size of the structure to *newsize*, where *newsize* is a value that is less than the value of SIZE specified on the CFRM policy for the structure, but greater than the current coupling facility size.

You can monitor the use of a coupling facility structure with the MQSC `DISPLAY CFSTATUS` command.

If no action is taken and a queue structure fills up, an MQRC\_STORAGE\_MEDIUM\_FULL return code is returned to the application. If the administration structure becomes full, the exact symptoms depend on which processes experience the error, but they might include the following problems:

- No responses to commands.
- Queue manager failure as a result of problems during commit processing.

### The CSQSYSAPPL structure

The *qsg-name*CSQSYSAPPL structure is an application structure for system queues. [Table 3](#) demonstrates an example of how to estimate the message data sizes for the default queues defined on the *qsg-name*CSQSYSAPPL structure.

<i>Table 24. Table showing CSQSYSAPPL usage against sizing.</i>	
<i>qsg-name</i> CSQSYSAPPL usage	Sizing
SYSTEM.QSG.CHANNEL.SYNCQ	2 messages of 500 bytes per active instance of a shared channel
SYSTEM.QSG.UR.RESOLUTION.QUEUE	1000 messages of 2 KB

The suggested initial structure definition values are as follows:

```
STRUCTURE NAME(qsg-nameCSQSYSAPPL)
INITSIZE(20480) /* 20 MB */
SIZE(30720) /* 30 MB */
FULLTHRESHOLD(85)
```

These values can be adjusted depending on your use of shared channels and group units of recovery.

## Mapping shared queues to structures

To define an application structure to IBM MQ, use the [DEFINE CFSTRUCT](#) command. When you define a structure to IBM MQ, do not include the QSG name prefix in the structure name. For example, to define an application structure to IBM MQ that has the name *qsg-name*APPLICATION1 in the CFRM policy, issue the following command:

```
DEFINE CFSTRUCT(APPLICATION1)
```

The CFSTRUCT attribute of the queue definition is used to map the queue to a structure. Specify the name of the CF structure without the QSG name prefix in this attribute. For example, the following command defines a shared queue on the APPLICATION1 structure:

```
DEFINE QLOCAL(myqueue) QSGDISP(SHARED) CFSTRUCT(APPLICATION1)
```

## **Planning your shared message data set (SMDS) environment**

If you are using queue sharing groups with SMDS offloading, IBM MQ needs to connect to a group of shared message data sets. Use this topic to help understand the data set requirements, and configuration required to store IBM MQ message data.

A *shared message data set* (described by the keyword SMDS) is a data set used by a queue manager to store offloaded message data for shared messages stored in a coupling facility structure.

**Note:** When defining SMDS data sets for a structure, you must have one for each queue manager.

When this form of data offloading is enabled, the **CFSTRUCT** requires an associated group of shared message data sets, one data set for each queue manager in the queue sharing group. The group of shared message data sets is defined to IBM MQ using the **DSGROUP** parameter on the **CFSTRUCT** definition. Additional parameters can be used to supply further optional information, such as the number of buffers to use and expansion attributes for the data sets.

Each queue manager can write to the data set which it owns, to store shared message data for messages written through that queue manager, and can read all of the data sets in the group.

A list describing the status and attributes for each data set associated with the structure is maintained internally as part of the **CFSTRUCT** definition, so each queue manager can check the definition to find out which data sets are currently available.

This data set information can be displayed using the **DISPLAY CFSTATUS TYPE(SMDS)** command to display current status and availability, and the **DISPLAY SMDS** command to display the parameter settings for the data sets associated with a specified **CFSTRUCT**.

Individual shared message data sets are effectively identified by the combination of the owning queue manager name (usually specified using the **SMDS** keyword) and the **CFSTRUCT** structure name.

This section describes the following topics:

- [The DSGROUP parameter](#)
- [The DSBLOCK parameter](#)
- [Shared message data set characteristics](#)
- [Shared message data set space management](#)
- [Access to shared message data sets](#)
- [Creating a shared message data set](#)
- [Shared message data set performance and capacity considerations](#)
- [Activating a shared message data set](#)

See [DEFINE CFSTRUCT](#) for details of these parameters.

For information on managing your shared message data sets, see [Managing shared message data sets](#) for further details.

## The DSGROUP parameter

The **DSGROUP** parameter on the **CFSTRUCT** definition identifies the group of data sets in which large messages for that structure are to be stored. Additional parameters may be used to specify the logical block size to be used for space allocation purposes and values for the buffer pool size and automatic data set expansion options.

The **DSGROUP** parameter must be set up before offloading to data sets can be enabled.

- If a new **CFSTRUCT** is being defined at **CFLEVEL (5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command.
- If an existing **CFSTRUCT** is being altered to increase the **CFLEVEL** to **CFLEVEL (5)** and the option **OFFLOAD(SMDS)** is specified or assumed, then the **DSGROUP** parameter must be specified on the same command if it is not already set.

## The DSBLOCK parameter

Space within each data set is allocated to queues as logical blocks of a fixed size (usually 256 KB) specified using the **DSBLOCK** parameter on the **CFSTRUCT** definition, then allocated to individual messages as ranges of pages of 4 KB (corresponding to the physical block size and control interval size) within each logical block. The logical block size also determines the maximum amount of message data that can be read or written in a single I/O operation, which is the same as the buffer size for the SMDS buffer pool.

A larger value of the **DSBLOCK** parameter can improve performance for very large messages by reducing the number of separate I/O operations. However, a smaller value decreases the amount of buffer storage required for each active request. The default value for the **DSBLOCK** parameter is 256 KB, which provides a reasonable balance between these requirements, so specifying this parameter might not normally be necessary.

## Shared message data set characteristics

A shared message data set is defined as a VSAM linear data set (LDS). Each offloaded message is stored in one or more blocks in the data set. The stored data is addressed directly by information in the coupling facility entries, like an extended form of virtual storage. There is no separate index or similar control information stored in the data set itself.

The direct addressing scheme means that for messages which fit into one block, only a single I/O operation is needed to read or write the block. When a message spans more than one block, the I/O operations for each block can be fully overlapped to minimize elapsed time, provided that sufficient buffers are available.

The shared message data set also contains a small amount of general control information, consisting of a header in the first page, which includes recovery and restart status information, and a space map checkpoint area which is used to save the free block space map at queue manager normal termination.

## Shared message data set space management

As background information for capacity, performance and operational considerations, it might be useful to understand the concepts of how space in shared message data sets is managed by the queue managers.

Free space in each shared message data set is tracked by its owning queue manager using a space map which indicates the number of pages in use within each logical block. The space map is maintained in main storage while the data set is open and saved in the data set when it is closed normally. (In recovery situations the space map is automatically rebuilt by scanning the messages in the coupling facility structure to find out which data set pages are currently in use).

When a shared message with offloaded message data is being written, the queue manager allocates a range of pages for each message block. If there is a partly used current logical block for the specified queue, the queue manager allocates space starting at the next free page in that block, otherwise it allocates a new logical block. If the whole message does not fit within the current logical block, the queue

manager splits the message data at the end of the logical block and allocates a new logical block for the next message block. This is repeated until space has been allocated for the whole message. Any unused space in the last logical block is saved as the new current logical block for the queue. When the data set is closed normally, any unused pages in current logical blocks are returned to the space map before it is saved.

When a shared message with offloaded message data has been read and is ready to be deleted, the queue manager processes the delete request by transferring the coupling facility entry for the message to a clean-up list monitored by the owning queue manager (which may be the same queue manager). When entries arrive on this list, the owning queue manager reads and deletes the entries and returns the freed ranges of pages to the space map. When all used pages in a logical block have been freed the block becomes available for reuse.

## Access to shared message data sets

Each shared message data set must be on shared direct access storage which is accessible to all queue managers in the queue sharing group.

During normal running, each queue manager opens its own shared message data set for read/write access, and opens any active shared message data sets for other queue managers for read-only access, so it can read messages stored by those queue managers. This means that each queue manager userid requires at least UPDATE access to its own shared message data set and READ access to all other shared message data sets for the structure.

If it is necessary to recover shared message data sets using **RECOVER CFSTRUCT**, the recovery process can be executed from any queue manager in the queue sharing group. A queue manager which may be used to perform recovery processing requires UPDATE access to all data sets that it may need to recover

## Creating a shared message data set

Each shared message data set should normally be created before the corresponding **CFSTRUCT** definition is created or altered to enable the use of this form of message offloading, as the **CFSTRUCT** definition changes will normally take effect immediately, and the data set will be required as soon as a queue manager attempts to access a shared queue which has been assigned to that structure. A sample job to allocate and pre-format a shared message data set is provided in SCSQPROC(CSQ4SMDS). The job must be customized and run to allocate a shared message data set for each queue manager which uses a CFSTRUCT with OFFLOAD(SMDS).

If the queue manager finds that offload support has been enabled and tries to open its shared message data set but it has not yet been created, the shared message data set will be flagged as unavailable. The queue manager will then be unable to store any large messages until the data set has been created and the queue manager has been notified to try again, for example using the **START SMDSCONN** command.

A shared message data set is created as a VSAM linear data set using an Access Method Services **DEFINE CLUSTER** command. The definition must specify **SHAREOPTIONS(2 3)** to allow one queue manager to open it for write access and any number of queue managers to read it at the same time. The default control interval size of 4 KB must be used. If the data set may need to expand beyond 4 GB, it must be defined using an SMS data class which has the VSAM extended addressability attribute. A shared message data set is eligible to reside in the extended addressing space (EAS) part of an extended address volumes (EAV).

Each shared message data set can either be empty or pre-formatted to binary zeros (using **CSQJUFMT** or a similar utility such as the sample job SCSQPROC(CSQ4SMDS)), before its initial use. If it is empty or only partly formatted when it is opened, the queue manager automatically formats the remaining space to binary zeros.

## Shared message data set performance and capacity considerations

Each shared message data set is used to store offloaded data for shared messages written to the associated **CFSTRUCT** by the owning queue manager, from regions within the same system. Each message that is offloaded takes up to 768 bytes of CF storage, made up of 256 bytes for the entry

and 512 bytes for the two elements of header and descriptor. Each offloaded message is stored in one or more pages (physical blocks of size 4 KB) in the data set.

The data set space required for a given number of offloaded messages can therefore be estimated by rounding up the overall message size (including the descriptor) to the next multiple of 4 KB and then multiplying by the number of messages.

As for a page set, when a shared message data set is almost full, it can optionally be expanded automatically. The default behavior for this automatic expansion can be set using the **DSEXPAND** parameter on the **CFSTRUCT** definition. This setting can be overridden for each queue manager using the **DSEXPAND** parameter on the **ALTER SMDS** command. Automatic expansion is triggered when the data set reaches 90% full and more space is required. If expansion is allowed but an expansion attempt is rejected by VSAM because no secondary space allocation was specified when the data set was defined, expansion is retried using a secondary allocation of 20% of the current size of the data set.

Provided that the shared message data set is defined with the extended addressability attribute, the maximum size is only limited by VSAM considerations to a maximum of 16 TB or 59 volumes. This is significantly larger than the 64 GB maximum size of a local page set.

## Activating a shared message data set

When a queue manager has successfully connected to an application coupling facility structure, it checks whether that structure definition specifies offloading using an associated **DSGROUP** parameter. If so, the queue manager allocates and opens its own shared message data set for write access, then it opens for read access any existing shared message data sets owned by other queue managers.

When a shared message data set is opened for the first time (before it has been recorded as active within the queue sharing group), the first page will not yet contain a valid header. The queue manager fills in header information to identify the queue sharing group, the structure name and the owning queue manager.

After the header has been completed, the queue manager registers the new shared message data set as active and broadcasts an event to notify any other active queue managers about the new data set.

Every time a queue manager opens a shared message data set it validates the header information to ensure that the correct data set is still being used and that it has not been damaged.

## **Planning your Db2 environment**

If you are using queue sharing groups, IBM MQ needs to attach to a Db2 subsystem that is a member of a data sharing group. Use this topic to help understand the Db2 requirements used to hold IBM MQ data.

IBM MQ needs to know the name of the data sharing group that it is to connect to, and the name of a Db2 subsystem (or Db2 group) to connect to, to reach this data sharing group. These names are specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro (described in [Using CSQ6SYSP](#)).

Within the data sharing group, shared Db2 tables are used to hold:

- Configuration information for the queue sharing group.
- Properties of IBM MQ shared and group objects.
- Optionally, data relating to offloaded IBM MQ messages.

IBM MQ provides a single set of sample jobs for defining the necessary Db2 table spaces, tables, and indexes. These jobs make use of Universal Table Spaces (UTS). Earlier versions of the product had two sets of jobs, one for UTS, and one for older types of table space, which have been deprecated by the most recent versions of Db2.

IBM MQ can still be used with older types of table space, and this might be appropriate if you already have an existing queue sharing group. However, if you are creating a new queue sharing group, it should use UTS.

Db2 V12 [Function level 508](#) provides a non disruptive migration process for migrating multi-table table spaces to universal table spaces. You can use this approach to migrate the multi-table table spaces, used

by existing queue sharing groups, to universal table spaces without taking an outage of the whole queue sharing group.

In Db2 V13, use the MOVE TABLE option of the ALTER TABLESPACE statement. See [Moving tables from multi-table table spaces to partition-by-growth table spaces](#) for more information.

By default Db2 uses the user ID of the person running the jobs as the owner of the Db2 resources. If this user ID is deleted then the resources associated with it are deleted, and so the table is deleted. Consider using a group ID to own the tables, rather than an individual user ID. You can do this by adding GROUP=groupname onto the JOB card, and specifying SET CURRENT SQLID='groupname' before any SQL statements.

IBM MQ uses the RRS Attach facility of Db2. This means that you can specify the name of a Db2 group that you want to connect to. The advantage of connecting to a Db2 group attach name (rather than a specific Db2 subsystem), is that IBM MQ can connect (or reconnect) to any available Db2 subsystem on the z/OS image that is a member of that group. There must be a Db2 subsystem that is a member of the data sharing group active on each z/OS image where you are going to run a queue-sharing IBM MQ subsystem, and RRS must be active.

## Db2 storage

For most installations, the amount of Db2 storage required is about 20 or 30 cylinders on a 3390 device. However, if you want to calculate your storage requirement, the following table gives some information to help you determine how much storage Db2 requires for the IBM MQ data. The table describes the length of each Db2 row, and when each row is added to or deleted from the relevant Db2 table. Use this information together with the information about calculating the space requirements for the Db2 tables and their indexes in the *Db2 for z/OS Installation Guide*.

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_QSG	252 bytes	A queue sharing group is added to the table with the ADD QSG function of the CSQ5PQSG utility.	A queue sharing group is removed from the table with the REMOVE QSG function of the CSQ5PQSG utility. (All rows relating to this queue sharing group are deleted automatically from all the other Db2 tables when the queue sharing group record is deleted.)
CSQ.ADMIN_B_QMGR	Up to 3828 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_STRUCTURE	1454 bytes	The first local queue definition, specifying the QSGDISP(SHARED) attribute, that names a previously unknown structure within the queue sharing group is defined.	The last local queue definition, specifying the QSGDISP(SHARED) attribute, that names a structure within the queue sharing group is deleted.
CSQ.ADMIN_B_SCST	342 bytes	A shared channel is started.	A shared channel becomes inactive.
CSQ.ADMIN_B_SSKT	254 bytes	A shared channel that has the NPMSPEED(NORMAL) attribute is started.	A shared channel that has the NPMSPEED(NORMAL) attribute becomes inactive.

Table 25. Planning your Db2 storage requirements (continued)

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_B_STRBACKUP	514 bytes	A new row is added to the CSQ.ADMIN_B_STRUCTURE table. Each entry is a dummy entry until the BACKUP CFSTRUCT command is run, which overwrites the dummy entries.	A row is deleted from the CSQ.ADMIN_B_STRUCTURE table.
CSQ.OBJ_B_AUTHINFO	3400 bytes	An authentication information object with QSGDISP(GROUP) is defined.	An authentication information object with QSGDISP(GROUP) is deleted.
CSQ.OBJ_B_QUEUE	Up to 3707 bytes	<ul style="list-style-type: none"> <li>• A queue with the QSGDISP(GROUP) attribute is defined.</li> <li>• A queue with the QSGDISP(SHARED) attribute is defined.</li> <li>• A model queue with the DEFTYPE(SHAREDYN) attribute is opened.</li> </ul>	<ul style="list-style-type: none"> <li>• A queue with the QSGDISP(GROUP) attribute is deleted.</li> <li>• A queue with the QSGDISP(SHARED) attribute is deleted.</li> <li>• A dynamic queue with the DEFTYPE(SHAREDYN) attribute is closed with the DELETE option.</li> </ul>
CSQ.OBJ_B_NAMELIST	Up to 15127 bytes	A namelist with the QSGDISP(GROUP) attribute is defined.	A namelist with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_CHANNEL	Up to 14127 bytes	A channel with the QSGDISP(GROUP) attribute is defined.	A channel with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_STGCLASS	Up to 2865 bytes	A storage class with the QSGDISP(GROUP) attribute is defined.	A storage class with the QSGDISP(GROUP) attribute class is deleted.
CSQ.OBJ_B_PROCESS	Up to 3347 bytes	A process with the QSGDISP(GROUP) attribute is defined.	A process with the QSGDISP(GROUP) attribute is deleted.
CSQ.OBJ_B_TOPIC	Up to 14520 bytes	A topic object with QSGDISP(GROUP) attribute is defined.	A topic object with QSGDISP(GROUP) attribute is deleted.
CSQ.EXTEND_B_QMGR	Less than 430 bytes	A queue manager is added to the table with the ADD QMGR function of the CSQ5PQSG utility.	A queue manager is removed from the table with the REMOVE QMGR function of the CSQ5PQSG utility.
CSQ.ADMIN_B_MESSAGES	87 bytes	For large message PUT (1 per BLOB).	For large message GET (1 per BLOB).

Table 25. Planning your Db2 storage requirements (continued)

Db2 table name	Length of row	A row is added when:	A row is deleted when:
CSQ.ADMIN_MSGS_BAUX1 CSQ.ADMIN_MSGS_BAUX2 CSQ.ADMIN_MSGS_BAUX3 CSQ.ADMIN_MSGS_BAUX4		These 4 tables contain message payload for large messages added into one of these 4 tables for each BLOB of the message. BLOBS are up to 511 KB in length, so if the message size is > 711 KB, there will be multiple BLOBs for this message.	

The use of large numbers of shared queue messages of size greater than 63 KB can have significant performance implications on your IBM MQ system. For more information, see SupportPac MP16, Capacity Planning and Tuning for IBM MQ for z/OS, at: [SupportPacs for IBM MQ and other project areas](#).

## Planning for backup and recovery

Developing backup and recovery procedures at your site is vital to avoid costly and time-consuming losses of data. IBM MQ provides means for recovering both queues and messages to their current state after a system failure.

This topic contains the following sections:

- [“Recovery procedures” on page 200](#)
- [“Tips for backup and recovery” on page 201](#)
- [“Recovering page sets” on page 203](#)
- [“Recovering CF structures” on page 204](#)
- [“Achieving specific recovery targets” on page 204](#)
- [“Backup considerations for other products” on page 206](#)
- [“Recovery and CICS” on page 206](#)
- [“Recovery and IMS” on page 207](#)
- [“Preparing for recovery on an alternative site” on page 207](#)
- [“Example of queue manager backup activity” on page 207](#)

### Recovery procedures

Develop the following procedures for IBM MQ:

- Creating a point of recovery.
- Backing up page sets.
- Backing up CF structures.
- Recovering page sets.
- Recovering from out-of-space conditions ( IBM MQ logs and page sets).
- Recovering CF structures.

See [Administration de IBM MQ for z/OS](#) for information about these.

Become familiar with the procedures used at your site for the following:

- Recovering from a hardware or power failure.
- Recovering from a z/OS component failure.



- Recovering from a site interruption, using off-site recovery.

## **Tips for backup and recovery**

Use this topic to understand some backup and recovery tasks.

The queue manager restart process recovers your data to a consistent state by applying log information to the page sets. If your page sets are damaged or unavailable, you can resolve the problem using your backup copies of your page sets (if all the logs are available). If your log data sets are damaged or unavailable, it might not be possible to recover completely.

Consider the following points:

- Periodically take backup copies
- Do not discard archive logs you might need
- Do not change the DDname to page set association

### **Periodically take backup copies**

A *point of recovery* is the term used to describe a set of backup copies of IBM MQ page sets and the corresponding log data sets required to recover these page sets. These backup copies provide a potential restart point in the event of page set loss (for example, page set I/O error). If you restart the queue manager using these backup copies, the data in IBM MQ is consistent up to the point that these copies were taken. Provided that all logs are available from this point, IBM MQ can be recovered to the point of failure.

The more recent your backup copies, the quicker IBM MQ can recover the data in the page sets. The recovery of the page sets is dependent on all the necessary log data sets being available.

In planning for recovery, you need to determine how often to take backup copies and how many complete backup cycles to keep. These values tell you how long you must keep your log data sets and backup copies of page sets for IBM MQ recovery.

When deciding how often to take backup copies, consider the time needed to recover a page set. The time needed is determined by the following:

- The amount of log to traverse.
- The time it takes an operator to mount and remove archive tape volumes.
- The time it takes to read the part of the log needed for recovery.
- The time needed to reprocess changed pages.
- The storage medium used for the backup copies.
- The method used to make and restore backup copies.

In general, the more frequently you make backup copies, the less time recovery takes, but the more time is spent making copies.

For each queue manager, you should take backup copies of the following:

- The archive log data sets
- The BSDS copies created at the time of the archive
- The page sets
- Your object definitions
- Your CF structures

To reduce the risk of your backup copies being lost or damaged, consider:

- Storing the backup copies on different storage volumes to the original copies.
- Storing the backup copies at a different site to the original copies.

- Making at least two copies of each backup of your page sets and, if you are using single logging or a single BSDS, two copies of your archive logs and BSDS. If you are using dual logging or BSDS, make a single copy of both archive logs or BSDS.

Before moving IBM MQ to a production environment, fully test and document your backup procedures.

### **Backing up your page sets**

You need to back up page sets regularly. Some enterprises back up the page sets twice a day.

You need the active and archive logs since a backup to be able to recover using the backup. You need enough log data to go back four checkpoints if the backup was taken when the queue manager was running.

You can use ADRDSSU FastReplication to back up page sets, and you can do this while the queue manager is active. Note that you need to ensure there is enough space in the storage pool.

### **Backing up your object definitions**

Create backup copies of your object definitions. To do this, use the MAKEDEF feature of the COMMAND function of the utility program (described in [Using the COMMAND function of CSQUTIL](#)).

You should do this whenever you take backup copies of your queue manager data sets, and keep the most current version.

### **Backing up your coupling facility structures**

If you have set up any queue sharing groups, even if you are not using them, you must take periodic backups of your CF structures. To do this, use the IBM MQ [BACKUP CFSTRUCT](#) command. You can use this command only on CF structures that are defined with the RECOVER(YES) attribute. If any CF entries for persistent shared messages refer to offloaded message data stored in a shared message data set (SMDS) or Db2, the offloaded data is retrieved and backed up together with the CF entries. Shared message data sets should not be backed up separately.

It is recommended that you take a backup of all your CF structures about every hour, to minimize the time it takes to restore a CF structure.

You could perform all your CF structure backups on a single queue manager, which has the advantage of limiting the increase in log use to a single queue manager. Alternatively, you could perform backups on all the queue managers in the queue sharing group, which has the advantage of spreading the workload across the queue sharing group. Whichever strategy you use, IBM MQ can locate the backup and perform a RECOVER CFSTRUCT from any queue manager in the queue sharing group. The logs of all the queue managers in the queue sharing group need to be accessed to recover the CF structure.

### **Backing up your message security policies**

If you are using Advanced Message Security to create a backup of your message security policies, create a backup using the [message security policy utility \(CSQUTIL\)](#) to run **dspmqspl** with the -export parameter, then save the policy definitions that are output to the EXPORT DD.

You should create a backup of your message security policies whenever you take backup copies of your queue manager data sets, and keep the most current version.

### **Do not discard archive logs you might need**

IBM MQ might need to use archive logs during restart. You must keep sufficient archive logs so that the system can be fully restored. IBM MQ might use an archive log to recover a page set from a restored backup copy. If you have discarded that archive log, IBM MQ cannot restore the page set to its current state. When and how you discard archive logs is described in [Discarding archive log data sets](#).

You can use the `/cpf DIS USAGE TYPE(ALL)` command to display the log RBA, and log range sequence number (LRSN) that you need to recover your queue manager's page sets and the queue sharing group's structures. You should then use the [print log map utility \(CSQJU004\)](#) to print bootstrap data set (BSDS) information for the queue manager to locate the logs containing the log RBA.

For CF structures, you need to run the CSQJU004 utility on each queue manager in the queue sharing group to locate the logs containing the LRSN. You need these logs and any later logs to be able to recover the page sets and structures.

## Do not change the DDname to page set association

IBM MQ associates page set number 00 with DDname CSQP0000, page set number 01 with DDname CSQP0001, and so on, up to CSQP0099. IBM MQ writes recovery log records for a page set based on the DDname that the page set is associated with. For this reason, you must not move page sets that have already been associated with a PSID DDname.

## Recovering page sets

Use this topic to understand the factors involved when recovering pages sets, and how to minimize restart times.

A key factor in recovery strategy concerns the time for which you can tolerate a queue manager outage. The total outage time might include the time taken to recover a page set from a backup, or to restart the queue manager after an abnormal termination. Factors affecting restart time include how frequently you back up your page sets, and how much data is written to the log between checkpoints.

To minimize the restart time after an abnormal termination, keep units of work short so that, at most, two active logs are used when the system restarts. For example, if you are designing an IBM MQ application, avoid placing an MQGET call that has a long wait interval between the first in-syncpoint MQI call and the commit point because this might result in a unit of work that has a long duration. Another common cause of long units of work is batch intervals of more than 5 minutes for the channel initiator.

You can use the [DISPLAY THREAD](#) command to display the RBA of units of work and to help resolve the old ones.

## How often must you back up a page set?

Frequent page set backup is essential if a reasonably short recovery time is required. This applies even when a page set is very small or there is a small amount of activity on queues in that page set.

If you use persistent messages in a page set, the backup frequency should be in hours rather than days. This is also the case for page set zero.

To calculate an approximate backup frequency, start by determining the target total recovery time. This consists of the following:

1. The time taken to react to the problem.
2. The time taken to restore the page set backup copy.

If you use SnapShot backup/restore, the time taken to perform this task is a few seconds. For information about SnapShot, see the *DFSMSdss Storage Administration Guide*.

3. The time the queue manager requires to restart, including the additional time needed to recover the page set.

This depends most significantly on the amount of log data that must be read from active and archive logs since that page set was last backed up. All such log data must be read, in addition to that directly associated with the damaged page set.

**Note:** When using *fuzzy backup* (where a snapshot is taken of the logs and page sets while a unit of work is active), it might be necessary to read up to three additional checkpoints, and this might result in the need to read one or more additional logs.

When deciding on how long to allow for the recovery of the page set, the factors that you need to consider are:

- The rate at which data is written to the active logs during normal processing depends on how messages arrive in your system, in addition to the message rate.

Messages received or sent over a channel result in more data logging than messages generated and retrieved locally.

- The rate at which data can be read from the archive and active logs.

When reading the logs, the achievable data rate depends on the devices used and the total load on your particular DASD subsystem.

With most tape units, it is possible to achieve higher data rates for archived logs with a large block size. However, if an archive log is required for recovery, all the data on the active logs must be read also.

## **Recovering CF structures**

Use this topic to understand the recovery process for CF structures.

At least one queue manager in the queue sharing group must be active to process a RECOVER CFSTRUCT command. CF structure recovery does not affect queue manager restart time, because recovery is performed by an already active queue manager.

The recovery process consists of two logical steps that are managed by the RECOVER CFSTRUCT command:

1. Locating and restoring the backup.
2. Merging all the logged updates to persistent messages that are held on the CF structure from the logs of all the queue managers in the queue sharing group that have used the CF structure, and applying the changes to the backup.

The second step is likely to take much longer because a lot of log data might need to be read. You can reduce the time taken if you take frequent backups, or if you recover multiple CF structures at the same time, or both.

The queue manager performing the recovery locates the relevant backups on all the other queue managers' logs using the data in Db2 and the bootstrap data sets. The queue manager replays these backups in the correct time sequence across the queue sharing group, from just before the last backup through to the point of failure.

The time it takes to recover a CF structure depends on the amount of recovery log data that must be replayed, which in turn depends on the frequency of the backups. In the worst case, it takes as long to read a queue manager's log as it did to write it. So if, for example, you have a queue sharing group containing six queue managers, an hour's worth of log activity could take six hours to replay. In general it takes less time than this, because reading can be done in bulk, and because the different queue manager's logs can be read in parallel. As a starting point, we recommend that you back up your CF structures every hour.

All queue managers can continue working with non-shared queues and queues in other CF structures while there is a failed CF structure. If the administration structure has also failed, at least one of the queue managers in the queue sharing group must be started before you can issue the RECOVER CFSTRUCT command.

Backing up CF structures can require considerable log writing capacity, and can therefore impose a large load on the queue manager doing the backup. Choose a lightly loaded queue manager for doing backups; for busy systems, add an additional queue manager to the queue sharing group and dedicate it exclusively for doing backups.

## **Achieving specific recovery targets**

Use this topic for guidance on how you can achieve specific recovery target times by adjusting backup frequency.

If you have specific recovery targets to achieve, for example, completion of the queue manager recovery and restart processing in addition to the normal startup time within xx seconds, you can use the following calculation to estimate your backup frequency (in hours):

$$\text{Backup frequency (in hours)} = \frac{\text{Required restart time (in secs)} * \text{System recovery log read rate (in MB/sec)}}{\text{Application log write rate (in MB/hour)}}$$

Formula (A)

**Note:** The examples given next are intended to highlight the need to back up your page sets frequently. The calculations assume that most log activity is derived from a large number of persistent messages. However, there are situations where the amount of log activity is not easily calculated. For example, in a queue sharing group environment, a unit of work in which shared queues are updated in addition to other resources might result in UOW records being written to the IBM MQ log. For this reason, the Application log write rate in Formula (A) can be derived accurately only from the observed rate at which the IBM MQ logs fill.

For example, consider a system in which IBM MQ MQI clients generate a total load of 100 persistent messages a second. In this case, all messages are generated locally.

If each message is of user length 1 KB, the amount of data logged each hour is approximately:

$$100 * (1 + 1.3) \text{ KB} * 3600 = \text{approximately } 800 \text{ MB}$$

where

- 100 = the message rate a second
- (1 + 1.3) KB = the amount of data logged for each 1 KB of persistent messages

Consider an overall target recovery time of 75 minutes. If you have allowed 15 minutes to react to the problem and restore the page set backup copy, queue manager recovery and restart must then complete within 60 minutes (3600 seconds) applying formula (A). Assuming that all required log data is on RVA2-T82 DASD, which has a recovery rate of approximately 2.7 MB a second, this necessitates a page set backup frequency of at least every:

$$3600 \text{ seconds} * 2.7 \text{ MB a second} / 800 \text{ MB an hour} = 12.15 \text{ hours}$$

If your IBM MQ application day lasts approximately 12 hours, one backup each day is appropriate. However, if the application day lasts 24 hours, two backups each day is more appropriate.

Another example might be a production system in which all the messages are for request-reply applications (that is, a persistent message is received on a receiver channel and a persistent reply message is generated and sent down a sender channel).

In this example, the achieved batch size is one, and so there is one batch for every message. If there are 50 request replies a second, the total load is 100 persistent messages a second. If each message is 1 KB in length, the amount of data logged each hour is approximately:

```
50((2 * (1+1.3) KB) + 1.4 KB + 2.5 KB) * 3600 = approximately 1500 MB
```

where:

```
50 = the message pair rate a second  
(2 * (1 + 1.3) KB) = the amount of data logged for each message pair  
1.4 KB = the overhead for each batch of messages  
received by each channel  
2.5 KB = the overhead for each batch of messages sent  
by each channel
```

To achieve the queue manager recovery and restart within 30 minutes (1800 seconds), again assuming that all required log data is on RVA2-T82 DASD, this requires that page set backup is carried out at least every:

```
1800 seconds * 2.7 MB a second / 1500 MB an hour = 3.24 hours
```

## Periodic review of backup frequency

Monitor your IBM MQ log usage in terms of MB an hour. Periodically perform this check and amend your page set backup frequency if necessary.

## Backup considerations for other products

If you are using IBM MQ with CICS or IMS then you must also consider the implications for your backup strategy with those products. The data facility hierarchical storage manager (DFHSM) manages data storage, and can interact with the storage used by IBM MQ.

## Backup and recovery with DFHSM

The data facility hierarchical storage manager (DFHSM) does automatic space-availability and data-availability management among storage devices in your system. If you use it, you need to know that it moves data to and from the IBM MQ storage automatically.

DFHSM manages your DASD space efficiently by moving data sets that have not been used recently to alternative storage. It also makes your data available for recovery by automatically copying new or changed data sets to tape or DASD backup volumes. It can delete data sets, or move them to another device. Its operations occur daily, at a specified time, and allow for keeping a data set for a predetermined period before deleting or moving it.

You can also perform all DFHSM operations manually. For more information on DFHSM, see the [z/OS DFSMS](#) product documentation. If you use DFHSM with IBM MQ, note that DFHSM does the following:

- Uses cataloged data sets.
- Operates on page sets and logs.
- Supports VSAM data sets.

## Recovery and CICS

The recovery of CICS resources is not affected by the presence of IBM MQ. CICS recognizes IBM MQ as a non-CICS resource (or external resource manager), and includes IBM MQ as a participant in any syncpoint coordination requests using the CICS resource manager interface (RMI). For more information about CICS recovery and the CICS resource manager interface, see the [CICS](#) product documentation.

## Recovery and IMS

IMS recognizes IBM MQ as an external subsystem and as a participant in syncpoint coordination. IMS recovery for external subsystem resources is described in the [IMS](#) product documentation.

### **Preparing for recovery on an alternative site**

If a total loss of an IBM MQ computing center, you can recover on another IBM MQ system at a recovery site.

To recover an IBM MQ system at a recovery site, you must regularly back up the page sets and the logs. As with all data recovery operations, the objectives of disaster recovery are to lose as little data, workload processing (updates), and time as possible.

At the recovery site:

- The recovery IBM MQ queue manager **must** have the same name as the lost queue manager.
- Ensure the system parameter module used on the recovery queue manager contains the same parameters as the lost queue manager.

See [Administering IBM MQ for z/OS](#) and [Troubleshooting IBM MQ for z/OS problems](#) for more information.

### **Example of queue manager backup activity**

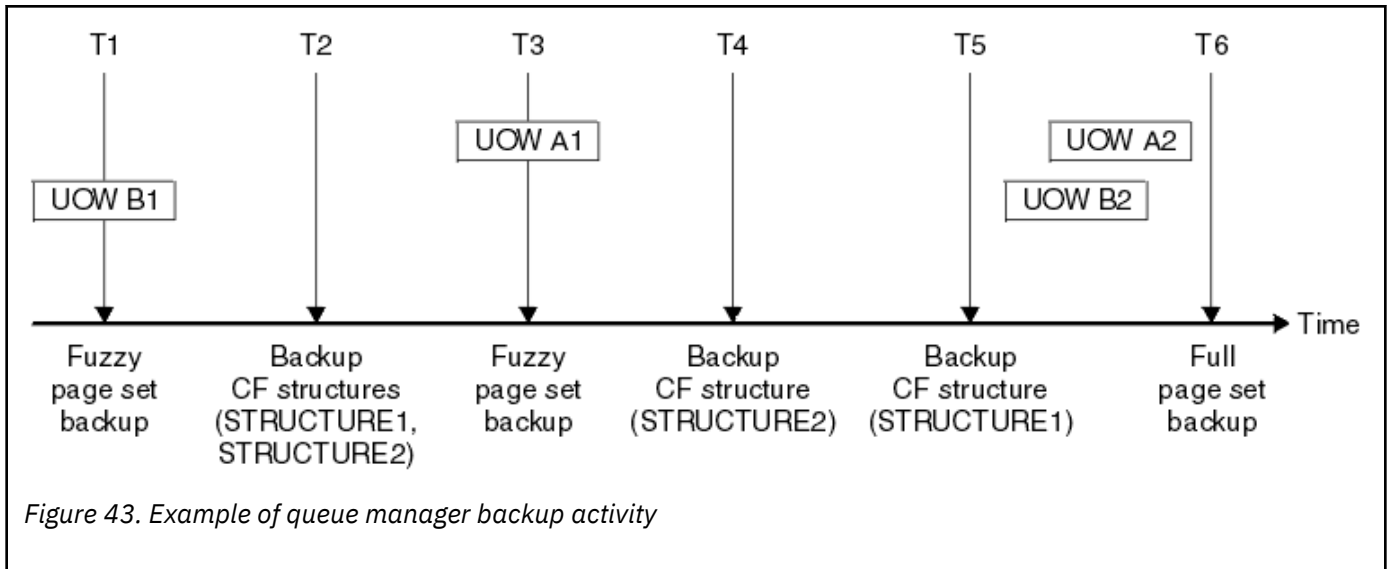
This topic shows as an example of queue manager backup activity.

When you plan your queue manager backup strategy, a key consideration is retention of the correct amount of log data. [Managing the logs](#) describes how to determine which log data sets are required, by reference to the system recovery RBA of the queue manager. IBM MQ determines the system recovery RBA using information about the following:

- Currently active units of work.
- Page set updates that have not yet been flushed from the buffer pools to disk.
- CF structure backups, and whether this queue manager's log contains information required in any recovery operation using them.

You must retain sufficient log data to be able to perform media recovery. While the system recovery RBA increases over time, the amount of log data that must be retained only decreases when subsequent backups are taken. CF structure backups are managed by IBM MQ, and so are taken into account when reporting the system recovery RBA. This means that in practice, the amount of log data that must be retained only reduces when page set backups are taken.

Figure 43 on page 208 shows an example of the backup activity on a queue manager that is a member of a queue sharing group, how the recovery RBA varies with each backup, and how that affects the amount of log data that must be retained. In the example the queue manager uses local and shared resources: page sets, and two CF structures, STRUCTURE1 and STRUCTURE2.



This is what happens at each point in time:

**Point in time T1**

A fuzzy backup is created of your page sets, as described in [How to back up and recover page sets](#).

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF application structures. This relates to the recovery of backups of STRUCTURE1 and STRUCTURE2 created earlier.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWB1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

**Point in time T2**

Backups of the CF structures are created. CF structure STRUCTURE1 is backed up first, followed by STRUCTURE2.

The amount of log data that must be retained is unchanged, because the same data as determined from the system recovery RBA at T1 is still required to recover using the page set backups taken at T1.

**Point in time T3**

Another fuzzy backup is created.

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover CF structure STRUCTURE1, because STRUCTURE1 was backed up before STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager (UOWA1).

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the fuzzy backup process.

You can now reduce the log data retained, as determined by this new system recovery RBA.

**Point in time T4**

A backup is taken of CF structure STRUCTURE2. The recovery RBA for the recovery of the oldest required CF structure backup relates to the backup of CF structure STRUCTURE1, which was backed up at time T2.

The creation of this CF structure backup has no effect on the amount of log data that must be retained.



### Point in time T5

A backup is taken of CF structure STRUCTURE1. The recovery RBA for recovery of the oldest required CF structure backup now relates to recovery of CF structure STRUCTURE2, which was backed up at time T4.

The creation of this CF structure backup has no effect on amount of log data that must be retained.

### Point in time T6

A full backup is taken of your page sets as described in [How to back up and recover page sets](#).

The system recovery RBA of the queue manager is the lowest of the following:

- The recovery RBAs of the page sets being backed up at this point.
- The lowest recovery RBA required to recover the CF structures. This relates to recovery of CF structure STRUCTURE2.
- The recovery RBA for the oldest currently active unit of work within the queue manager. In this case, there are no current units of work.

The system recovery RBA for this point in time is given by messages issued by the DISPLAY USAGE command, which is part of the full backup process.

Again, the log data retained can be reduced, because the system recovery RBA associated with the full backup is more recent.

z/OS

## Planning your z/OS UNIX environment

Certain processes within the IBM MQ queue manager, channel initiator, and mqweb server use z/OS UNIX System Services (z/OS UNIX) for their normal processing.

The queue manager and channel initiator started task user IDs need an OMVS segment with a UID defined in order to be able to access z/OS UNIX. The user IDs require no special permissions in z/OS UNIX.

**Note:** Although the queue manager and channel initiator make use of z/OS UNIX facilities (for example, to interface with TCP/IP services), they do not need to access any of the content of the IBM MQ installation directory in the z/OS UNIX file system. As a result, the queue manager and channel initiator do not require any configuration to specify the path for the z/OS UNIX file system.

The mqweb server, which hosts the IBM MQ Console and REST API, makes use of files in the IBM MQ installation directory in the z/OS UNIX file system. It also needs access to another file system which is used to store data such as configuration and log files. The mqweb started task JCL needs to be customized to reference these z/OS UNIX file systems.

The content of the IBM MQ directory in the z/OS UNIX file system is also used by applications connecting to IBM MQ. For example, applications using the IBM MQ classes for Java or IBM MQ classes for JMS interfaces.

See the following topics for the relevant configuration instructions:

- [Environment variables relevant to IBM MQ classes for Java](#)
- [IBM MQ classes for Java libraries](#)
- [Setting environment variables](#)
- [Configuring the Java Native Interface \(JNI\) libraries](#)

z/OS

## Planning for Advanced Message Security

TLS (or SSL) can be used to encrypt and protect messages flowing on a network, but this does not protect messages when they are on a queue ("at rest"). Advanced Message Security (AMS) protects the messages from the time that they are first put to a queue, until they are got, so that only the intended recipients of the message can read that message. The messages are encrypted and signed during put processing, and unprotected during get processing.

AMS can be configured to protect messages in different ways:

1. A message can be signed. The message is in clear text, but there is a checksum, which is signed. This allows any changes in the message content to be detected. From the signed content, you can identify who signed the data.
2. A message can be encrypted. The contents are not visible to anyone without the decryption key. The decryption key is encrypted for each recipient.
3. A message can be encrypted and signed. The decryption key is encrypted for each recipient, and from the signing you can identify who sent the message.

The encryption and signing use digital certificates and key rings.

You can set up a client to use AMS, so the data is protected before the data is put on the client channel. Protected messages can be sent to a remote queue manager, and you need to configure the remote queue manager to process these messages.

## Setting up AMS

An AMS address space is used for doing the AMS work. This has additional security set up, to give access to and protect the use of key rings and certificates.

You configure which queues are to be protected by using a utility program (CSQOUTIL) to define the security policies for queues.

## Once AMS is set up

You need to set up a digital certificate and a key ring for people who put messages, and the people who get messages.

If a user, Alice, on z/OS needs to send a message to Bob, AMS needs a copy of the public certificate for Bob.

If Bob wants to process a message from Alice, AMS needs the public certificate for Alice, or the same certificate authority certificate used by Alice.



**Attention:** You need to:

- Carefully plan who can put to, or get from, queues
- Identify the people and their certificate names.

It is easy to make mistakes, and problems can be hard to resolve.

## Related concepts

[“Planning for your queue manager” on page 154](#)

When you are setting up a queue manager, your planning should allow for the queue manager to grow, so that the queue manager meets the needs of your enterprise.

z/OS

## Planning for Managed File Transfer

Use this section as guidance on how you need to set up your system to run Managed File Transfer (MFT) on z/OS.

z/OS

## Planning for Managed File Transfer - hardware and software requirements

Use this topic as guidance on how you need to set up hardware and software requirements on your system to run Managed File Transfer (MFT) on z/OS.

## Software requirements

Managed File Transfer is written in Java, with some shell scripts and JCL to configure and operate the program.

**Important:** You must be familiar with z/OS UNIX System Services (z/OS UNIX) in order to configure Managed File Transfer. For example:

- The file directory structure, with names such as `/u/userID/myfile.txt`
- z/OS UNIX commands, for example:
  - `cd` (change directory)
  - `ls` (list)
  - `chmod` (change the file permissions)
  - `chown` (change file ownership or groups which can access the file or directory)

You require the following products in z/OS UNIX to be able to configure and run MFT:

1. Java, for example, in directory `/java/java80_bit64_GA/J8.0_64/`
2. IBM MQ 9.4.0, for example, in directory `/mqm/V9R3M0`
3. If you want to use Db2 for status and history, you need to install Db2 JDBC libraries, for example, in directory `/db2/db2v10/jdbc/libs`.

## Product registration

At startup Managed File Transfer checks the registration in `sys1.parmlib(IFAPRDxx)` concatenation. The following code is an example of how you register MFT:

```
PRODUCT OWNER('IBM CORP')
NAME('WS MQ FILE TRANS')
ID(5655-MFT)
VERSION(*) RELEASE(*) MOD(*)
FEATURENAME('WS MQ FILE TRANS')
STATE(ENABLED)
```

## Disk space

The IBM MQ for z/OS Program Directory states the DASD and zFS storage requirements for Managed File Transfer. For download links for the Program Directory for IBM MQ for z/OS, see [IBM MQ 9.4 PDF files for product documentation and Program Directories](#).

## Planning for Managed File Transfer - topologies

Use this topic as guidance on what topology you need on your system to run Managed File Transfer (MFT) on z/OS.

### Managed File Transfer queue managers

IBM MQ Managed File Transfer topologies consist of:

#### Agents, and their associated queue managers

The agent uses system queues hosted on their agent queue manager to maintain state information and receive requests for work.

#### A command queue manager

This acts as a gateway into an MFT topology. It is connected to the agent queue managers through either sender and receiver channels, or clustering. When certain commands are run, they connect directly to the command queue manager, and send a message to the specified agent. This message is routed through the IBM MQ network to the agent queue manager, where it is picked up by the agent and processed.

#### A coordination queue manager

This is a central hub that has knowledge of the entire topology. The coordination queue manager is connected to all of the agent queue managers in a topology through either sender and receiver

channels, or using clustering. Agents regularly publish status information to the coordination queue manager, and store their transfer templates there.

It is possible for a single queue manager to perform multiple roles within a topology. For example, the same queue manager can be configured as both the coordination queue manager and the command queue manager for a topology.

If you are using multiple queue managers you need to set up channels between the queue managers. You can either do this by using clustering or by using point-to-point connections.

When using IBM MQ Managed File Transfer for z/OS, there are a number of things to consider when determining which queue managers to use for the different roles within a topology.

### Agent queue managers

The agent queue manager for an IBM MQ Managed File Transfer for z/OS agent must be running on z/OS.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.1 or later
- And, the agent queue manager is licensed for IBM MQ Advanced for z/OS Value Unit Edition (Advanced VUE)

the agent can connect to the queue manager using the CLIENT transport.



Figure 44. MFT 9.1 agents on z/OS can connect to a queue manager using the CLIENT transport, assuming the queue manager is licensed for Advanced VUE.

If:

- The agent is running Managed File Transfer for z/OS on IBM MQ 9.0 or earlier
- Or, the agent queue manager is running Managed File Transfer for z/OS on IBM MQ 9.0 or later, and the agent queue manager is licensed for either MFT, IBM MQ Advanced for z/OS, or Advanced VUE

the agent must connect to the queue manager using the BINDINGS transport.



Figure 45. MFT 9.0 agents on z/OS and 9.1 agents that have an agent queue manager licensed for either MFT or IBM MQ Advanced, must connect using the BINDINGS transport.

### Command queue managers

The [Which MFT commands and processes connect to which queue manager](#) topic shows all of the commands that connect to the command queue manager for a Managed File Transfer topology.

**Note:** When running these commands on z/OS, the command queue manager must also be on z/OS.

If the command queue manager is licensed for Advanced VUE, the commands can connect to the queue manager using the CLIENT transport. Otherwise, the commands must connect to the command queue manager using the BINDINGS transport.

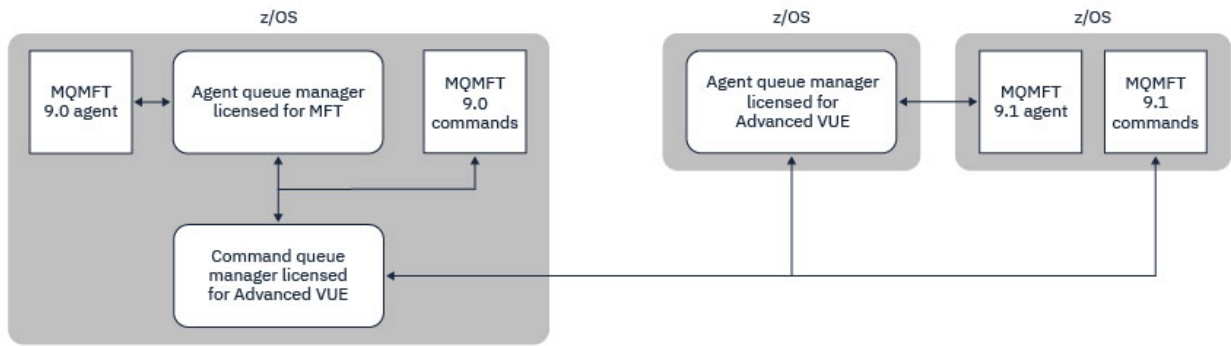


Figure 46. Commands connect to the command queue manager for an MFT topology. When running these commands on z/OS, the command queue manager must also be on z/OS

### Coordination queue managers

IBM MQ Managed File Transfer for z/OS agents can be part of a topology where the coordination queue manager is either running on z/OS, or is running on a multiplatform.

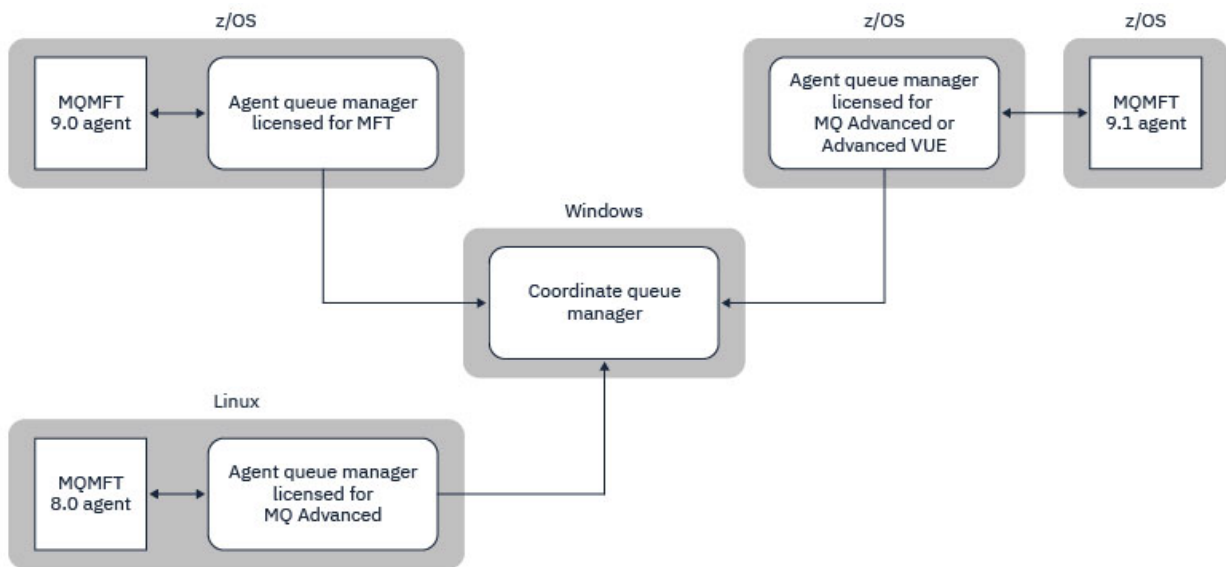


Figure 47. MFT agents running on z/OS can be part of an MFT topology where the coordination queue manager is running on an IBM MQ multiplatform.

The [Which MFT commands and processes connect to which queue manager](#) topic shows the commands that connect to the coordination queue manager for a Managed File Transfer topology. It is possible to run these commands on z/OS and have then connect to the coordination queue manager running on a different platform.

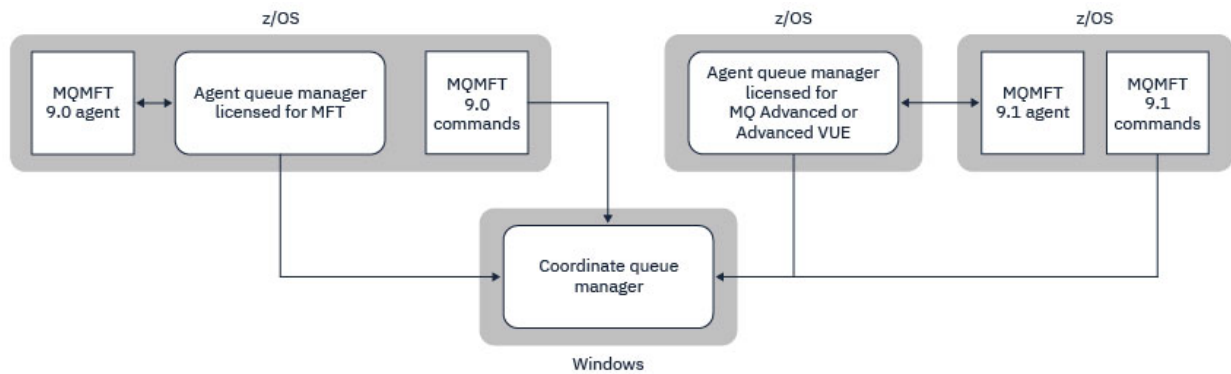


Figure 48. Certain commands, such as **fteListAgents**, connect directly to the coordination queue manager for an MFT topology.

### How many agents do I need?

The agents do the work in transferring data, and when you make a request to transfer data you specify the name of an agent.

By default an agent can process 25 send and 25 receive requests concurrently. You can configure these processes. See [Managed File Transfer configuration options on z/OS](#) for more information.

If the agent is busy then work is queued. The time taken to process a request depends on multiple factors, for example, the amount of data to be sent, the network bandwidth, and the delay on the network.

You might want to have multiple agents to process work in parallel.

You can also control which resources an agent can access, so you might want some agents to work with a limited subset of data.

If you want to process requests with different priority you can use multiple agents and use workload manager to set the priority of the jobs.

### Running the agents

Typically the agents are long running processes. The processes can be submitted as jobs that run in batch, or as started tasks.

## z/OS Planning for Managed File Transfer - security considerations

Use this topic as guidance on what security considerations you need on your system to run Managed File Transfer (MFT) on z/OS.

### Security

You need to identify which user IDs are going to be used for MFT configuration and for MFT operation.

You need to identify the files or queues you transfer, and which user IDs are going to be submitting transfer requests to MFT.

When you customize the agents and logger, you specify the group of users that is allowed to run MFT services, or do MFT administration.

You should set up this group before you start customizing MFT. As MFT uses IBM MQ queues, if you have security enabled in the queue manager, MFT requires access to the following resources:

Table 26. MQADMIN resource class	
Name	Access required
QUEUE.SYSTEM.FTE.EVENT.agent_name	Update

<i>Table 26. MQADMIN resource class (continued)</i>	
<b>Name</b>	<b>Access required</b>
QUEUE.SYSTEM.FTE.COMMAND.agent_name	Update
CONTEXT.SYSTEM.FTE.COMMAND.agent_name	Update
QUEUE.SYSTEM.FTE.STATE.agent_name	Update
QUEUE.SYSTEM.FTE.DATA.agent_name	Update
QUEUE.SYSTEM.FTE.REPLY.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHAGT1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHTRN1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHOPS1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHSCH1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHMON1.agent_name	Update
QUEUE.SYSTEM.FTE.AUTHADM1.agent_name	Update

<i>Table 27. MQQUEUE resource class</i>	
<b>Name</b>	<b>Access required</b>
SYSTEM.FTE.AUTHAGT1.agent_name	Update
SYSTEM.FTE.AUTHTRN1.agent_name	Update
SYSTEM.FTE.AUTHOPS1.agent_name	Update
SYSTEM.FTE.AUTHSCH1.agent_name	Update
SYSTEM.FTE.AUTHMON1.agent_name	Update

You can use user sandboxing to determine which parts of the file system the user who requests the transfer can access.

To enable user sandboxing, add the `userSandboxes=true` statement to the *agent.properties* file for the agent that you want to restrict, and add appropriate values to the `MQ_DATA_PATH/mqft/config/coordination_qmgr_name/agents/agent_name/UserSandboxes.xml` file.

See [Working with user sandboxes](#) for further information.

This user ID is configured in `UserSandboxes.xml` files.

This XML file has information like user ID, or user ID\* and a list of resource that can be used (included), or cannot be used (excluded). You need to define specific user IDs that can access which resources: for example:

<i>Table 28. Example user ID together with access to specific resources</i>			
<b>User ID</b>	<b>Access</b>	<b>Include or Exclude</b>	<b>Resource</b>
Admin*	Read	Include	/home/user/**
Admin*	Read	Exclude	/home/user/private/**
Sysprog	Read	Include	/home/user/**
Admin*	Read	Include	Application.reply.queue

## Notes:

1. If type=queue is specified, the resource is either a queue name, or queue@qmgr.
2. If the resource begins with //, the resource is a data set; otherwise the resource is a file in z/OS UNIX.
3. The user ID is the user ID from the MQMD structure, so this might not reflect the user ID that actually puts the message.
4. For requests on the local queue manager you can use MQADMIN CONTEXT.\* to limit which users can set this value.
5. For requests coming in over a remote queue manager, you have to assume that the distributed queue managers have security enabled to prevent unauthorized setting of the user ID in the MQMD structure.
6. A user ID of SYSPROG1 on a Linux machine, is the same user ID SYSPROG1 for the security checking on z/OS.

z/OS

## Planning to use the IBM MQ Console and REST API on z/OS

The IBM MQ Console and REST API are applications that run in a WebSphere Liberty (Liberty) server known as mqweb. The mqweb server runs as a started task. The IBM MQ Console allows a web browser to be used to administer queue managers. The REST API provides a simple programmatic interface for applications to do queue manager administration, and to perform messaging.

### Installation and configuration files

You need to install the IBM MQ for z/OS UNIX System Services Web Components feature, which will install the files needed to run the mqweb server in z/OS UNIX System Services (z/OS UNIX). You need to be familiar with z/OS UNIX to be able to configure and manage the mqweb server.

See [IBM MQ for z/OS Program Directory PDF files](#) for information on installing IBM MQ for z/OS UNIX System Services Components.

The IBM MQ files in z/OS UNIX are installed with various attributes set that are required for the correct operation of the mqweb server. If you need to copy the IBM MQ z/OS UNIX installation files, for example if you have installed IBM MQ on one system, and run IBM MQ on a different system, you should copy the IBM MQ ZFS created during the installation, and mount it read only at the destination. Copying the files in other ways might cause some file attributes to be lost.

You need to decide upon the location for, and create, a Liberty user directory when you create the mqweb server. This directory contains configuration and log files, and the location can be something similar to /var/mqm/mqweb.

### Using the IBM MQ Console and REST API with queue managers at different levels

The REST API can directly interact only with queue managers that run at the same Version, Release, and Modification (VRM) as the mqweb server which runs the REST API. For example, the IBM MQ 9.4.0 REST API can directly interact only with local queue managers at IBM MQ 9.4.0, and the IBM MQ 9.3.5 REST API can directly interact only with local queue managers at IBM MQ 9.3.5.

You can use the REST API to administer a queue manager at a different version from the mqweb server by configuring a gateway queue manager. However, you need at least one queue manager at the same version as the mqweb server to act as the gateway queue manager. For more information, see [Remote administration using the REST API](#).

The IBM MQ Console can be used to manage local queue managers that run at the same version as the IBM MQ Console. From IBM MQ 9.3.0, you can also use the IBM MQ Console to administer a queue manager running on a remote system, or at a different version to the IBM MQ Console. For more information, see [IBM MQ Console: Adding a remote queue manager](#).



## Migration

If you have only one queue manager, you can run the mqweb server as a single started task, and change the libraries it uses when you migrate your queue manager.

If you have more than one queue manager, during migration you can start mqweb servers at different versions by using started tasks with different names. These names can be any name you want. For example, you can start an IBM MQ 9.3.0 mqweb server using a started task named MQWB0930, and an IBM MQ 9.3.5 mqweb server using a started task named MQWB0935.

Then, when you migrate the queue managers from one version to a later version, the queue managers become available in the mqweb server for the later version, and are no longer available in the mqweb server for the earlier version.

After you have migrated all the queue managers to the later version, you can delete the mqweb server for the earlier version.

## HTTP ports

The mqweb server uses up to two ports for HTTP:

- One for HTTPS, with a default value of 9443.
- One for HTTP. HTTP is not enabled by default, but if enabled, has a default value of 9080.

If the default port values are in use, you must allocate other ports. If you have more than one mqweb server running simultaneously for more than one version of IBM MQ, you must allocate separate ports for each version. For more information on setting the ports that the mqweb server uses, see [Configuring the HTTP and HTTPS ports](#).

You can use the following TSO command to display information about a port:

```
NETSTAT TCP tcpip (PORT portNumber)
```

where *tcpip* is the name of the TCP/IP address space, and *portNumber* specifies the number of the port to display information about.

## Security - starting the mqweb server

The mqweb server user ID needs certain authorities. For more information, see [Authority required by the mqweb server started task user ID](#).

## Security - using the IBM MQ Console and REST API

When you use the IBM MQ Console and REST API, you must authenticate as a user that is included in a configured registry. These users are assigned specific roles that determine the actions the users can perform. For example, to use the messaging REST API, a user must be assigned the MQWebUser1 role. For more information about the available roles for the IBM MQ Console and REST API, and the access that these roles grant, see [Roles on the IBM MQ Console and REST API](#).

For more information about configuring security for the IBM MQ Console and REST API, see [IBM MQ Console and REST API security](#).



## Remarques

---

:NONE.

Le présent document peut contenir des informations ou des références concernant certains produits, logiciels ou services IBM non annoncés dans ce pays. Consultez votre représentant IBM local pour obtenir des informations sur les produits et services actuellement disponibles dans votre région. Toute référence à un produit, logiciel ou service IBM n'implique pas que seul ce produit, logiciel ou service IBM puisse être utilisé. Tout produit, programme ou service fonctionnellement équivalent qui ne porte pas atteinte à un droit de propriété intellectuelle IBM peut être utilisé à la place. Il est de la responsabilité de l'utilisateur d'évaluer et de vérifier lui-même les installations et applications réalisées avec des produits, logiciels ou services non expressément référencés par IBM.

IBM peut détenir des brevets ou des demandes de brevet couvrant les produits mentionnés dans le présent document. La remise de ce document ne vous donne aucun droit de licence sur ces brevets ou demandes de brevet. Si vous désirez recevoir des informations concernant l'acquisition de licences, veuillez en faire la demande par écrit à l'adresse suivante :

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

Pour obtenir des informations sur les licences relatives aux informations sur deux octets (DBCS), contactez le service de la propriété intellectuelle IBM de votre pays ou envoyez vos demandes de renseignements, par écrit, à :

Intellectual Property Licensing  
Legal and Intellectual Property Law  
IBM Japan, Ltd.  
19-21, Nihonbashi-Hakozakicho, Chuo-ku  
Tokyo 103-8510, Japan

**Le paragraphe suivant ne s'applique ni au Royaume-Uni, ni dans aucun pays dans lequel il serait contraire aux lois locales.** LE PRESENT DOCUMENT EST LIVRE "EN L'ETAT" SANS AUCUNE GARANTIE EXPLICITE OU IMPLICITE. IBM DECLINE NOTAMMENT TOUTE RESPONSABILITE RELATIVE A CES INFORMATIONS EN CAS DE CONTREFACON AINSI QU'EN CAS DE DEFAUT D'APTITUDE A L'EXECUTION D'UN TRAVAIL DONNE. Certaines juridictions n'autorisent pas l'exclusion des garanties implicites, auquel cas l'exclusion ci-dessus ne vous sera pas applicable.

Le présent document peut contenir des inexactitudes ou des coquilles. Ce document est mis à jour périodiquement. Chaque nouvelle édition inclut les mises à jour. IBM peut, à tout moment et sans préavis, modifier les produits et/ou programmes décrits dans ce document.

Les références à des sites Web non IBM sont fournies uniquement à titre d'information et n'impliquent en aucune façon une adhésion de ces sites Web. Les éléments figurant sur ces sites Web ne font pas partie des éléments du présent produit IBM et l'utilisation de ces sites relève de votre seule responsabilité.

IBM pourra utiliser ou diffuser, de toute manière qu'elle jugera appropriée et sans aucune obligation de sa part, tout ou partie des informations qui lui seront fournies.

Les licenciés souhaitant obtenir des informations permettant : (i) l'échange des données entre des logiciels créés de façon indépendante et d'autres logiciels (dont celui-ci), et (ii) l'utilisation mutuelle des données ainsi échangées, doivent adresser leur demande à :

IBM Corporation  
Coordinateur d'interopérabilité logicielle, département 49XA  
3605 Autoroute 52 N

Rochester, MN 55901  
U.S.A.

Ces informations peuvent être soumises à des conditions particulières, prévoyant notamment le paiement d'une redevance.

Le logiciel sous licence décrit dans le présent document et tous les éléments sous disponibles s'y rapportant sont fournis par IBM conformément aux dispositions du Contrat sur les produits et services IBM, aux Conditions Internationales d'Utilisation de Logiciels IBM ou de tout autre accord équivalent.

Les données de performance indiquées dans ce document ont été déterminées dans un environnement contrôlé. Par conséquent, les résultats peuvent varier de manière significative selon l'environnement d'exploitation utilisé. Certaines mesures évaluées sur des systèmes en cours de développement ne sont pas garanties sur tous les systèmes disponibles. En outre, elles peuvent résulter d'extrapolations. Les résultats peuvent donc varier. Il incombe aux utilisateurs de ce document de vérifier si ces données sont applicables à leur environnement d'exploitation.

Les informations concernant des produits non IBM ont été obtenues auprès des fournisseurs de ces produits, par l'intermédiaire d'annonces publiques ou via d'autres sources disponibles. IBM n'a pas testé ces produits et ne peut confirmer l'exactitude de leurs performances ni leur compatibilité. Elle ne peut recevoir aucune réclamation concernant des produits non IBM. Toute question concernant les performances de produits non IBM doit être adressée aux fournisseurs de ces produits.

Toute instruction relative aux intentions d'IBM pour ses opérations à venir est susceptible d'être modifiée ou annulée sans préavis, et doit être considérée uniquement comme un objectif.

Le présent document peut contenir des exemples de données et de rapports utilisés couramment dans l'environnement professionnel. Ces exemples mentionnent des noms fictifs de personnes, de sociétés, de marques ou de produits à des fins illustratives ou explicatives uniquement. Toute ressemblance avec des noms de personnes, de sociétés ou des données réelles serait purement fortuite.

Licence sur les droits d'auteur :

Le présent logiciel contient des exemples de programmes d'application en langage source destinés à illustrer les techniques de programmation sur différentes plateformes d'exploitation. Vous avez le droit de copier, de modifier et de distribuer ces exemples de programmes sous quelque forme que ce soit et sans paiement d'aucune redevance à IBM, à des fins de développement, d'utilisation, de vente ou de distribution de programmes d'application conformes aux interfaces de programmation des plateformes pour lesquels ils ont été écrits ou aux interfaces de programmation IBM. Ces exemples de programmes n'ont pas été rigoureusement testés dans toutes les conditions. Par conséquent, IBM ne peut garantir expressément ou implicitement la fiabilité, la maintenabilité ou le fonctionnement de ces programmes.

Si vous visualisez ces informations en ligne, il se peut que les photographies et illustrations en couleur n'apparaissent pas à l'écran.

## Documentation sur l'interface de programmation

---

Les informations d'interface de programmation, si elles sont fournies, sont destinées à vous aider à créer un logiciel d'application à utiliser avec ce programme.

Ce manuel contient des informations sur les interfaces de programmation prévues qui permettent au client d'écrire des programmes pour obtenir les services d'IBM MQ.

Toutefois, lesdites informations peuvent également contenir des données de diagnostic, de modification et d'optimisation. Ces données vous permettent de déboguer votre application.

**Important :** N'utilisez pas ces informations de diagnostic, de modification et d'optimisation en tant qu'interface de programmation car elles sont susceptibles d'être modifiées.

## Marques

---

IBM, le logo IBM, ibm.com, sont des marques d'IBM Corporation dans de nombreux pays. La liste actualisée de toutes les marques d'IBM est disponible sur la page Web "Copyright and trademark".

information"www.ibm.com/legal/copytrade.shtml. Les autres noms de produits et de services peuvent être des marques d'IBM ou d'autres sociétés.

Microsoft et Windows sont des marques de Microsoft Corporation aux Etats-Unis et/ou dans certains autres pays.

UNIX est une marque de The Open Group aux Etats-Unis et dans certains autres pays.

Linux est une marque de Linus Torvalds aux Etats-Unis et/ou dans certains autres pays.

Ce produit inclut des logiciels développés par le projet Eclipse (<https://www.eclipse.org/>).

Java ainsi que tous les logos et toutes les marques incluant Java sont des marques d'Oracle et/ou de ses sociétés affiliées.







Référence :

(1P) P/N: