

9.4

*Surveillance et performances 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 403.

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>Contrôle et performances.....</b>	<b>5</b>
Surveillance de votre réseau IBM MQ.....	5
Intégration d' OpenTelemetry.....	5
Comportement du diagnostic d'intégrité du gestionnaire de files d'attente.....	6
Moniteur d'événements.....	9
Surveillance de message.....	62
Messages de comptabilité et de statistiques.....	143
Trace de l'activité d'application.....	212
Rubriques système pour la surveillance et la trace de l'activité.....	301
Surveillance en temps réel.....	312
Surveillance des clusters.....	325
Surveillance de l'équilibrage des applications.....	328
Monitoring performance and resource usage on z/OS.....	331
Optimisation de votre réseau IBM MQ.....	387
Réglage des canaux de connexion client et serveur.....	387
Optimisation des réseaux de publication / abonnement distribués.....	389
Réduction du nombre de rubriques non souhaitées dans l'arborescence de rubrique.....	399
Aspera gateway peut améliorer les performances sur les réseaux à haute latence.....	401
<b>Remarques.....</b>	<b>403</b>
Documentation sur l'interface de programmation.....	404
Marques.....	404



# IBM MQ Contrôle et performances

---

Utilisez les informations de surveillance et les conseils de cette section, ainsi que les conseils d'optimisation spécifiques, pour améliorer les performances de votre réseau de gestionnaires de files d'attente.

## Pourquoi et quand exécuter cette tâche

En fonction de la taille et de la complexité de votre réseau de gestionnaires de files d'attente, vous pouvez obtenir une série d'informations à partir de la surveillance du réseau. Vous pouvez utiliser ces informations, ainsi que les informations fournies dans des conseils d'optimisation spécifiques, pour vous aider à optimiser les performances de votre réseau.

## Surveillance de votre réseau IBM MQ

---

Un certain nombre de techniques de surveillance sont disponibles dans IBM MQ pour obtenir des statistiques et d'autres informations spécifiques sur l'exécution de votre réseau de gestionnaires de files d'attente. Utilisez les informations de surveillance et les conseils de cette section pour améliorer les performances de votre réseau de gestionnaires de files d'attente.

La liste suivante fournit des exemples de raisons de surveillance de votre réseau de gestionnaires de files d'attente:

- Détectez les problèmes dans votre réseau de gestionnaires de files d'attente.
- Aide à déterminer les causes des problèmes dans votre réseau de gestionnaires de files d'attente.
- Améliorez l'efficacité de votre réseau de gestionnaires de files d'attente.
- Familiarisez-vous avec l'exécution de votre réseau de gestionnaires de files d'attente.
- Vérifiez que votre réseau de gestionnaires de files d'attente s'exécute correctement.
- Générer des messages lorsque certains événements se produisent.
- Enregistrez l'activité des messages.
- Déterminez le dernier emplacement connu d'un message.
- Vérifiez les différentes statistiques d'un réseau de gestionnaires de files d'attente en temps réel.
- Générez une trace d'audit.
- Compte pour l'utilisation des ressources d'application.
- Planification des capacités

V 9.4.0

ALW

## Intégration d' OpenTelemetry

Vous pouvez intégrer IBM MQ à un système de traçage OpenTelemetry .

### Fonction de trace OpenTelemetry

La fonction de trace d' OpenTelemetry vous permet d'observer comment les applications se comportent dans un flux de données. Le flux de données peut, et incorpore souvent, plusieurs applications différentes. Le traçage peut vous montrer tout le parcours et fournir des informations sur le comportement de chaque application individuelle. IBM MQ fournit un service de traçage qui vous permet de l'intégrer à un système de traçage OpenTelemetry .

Le service de traçage IBM MQ OpenTelemetry est implémenté en tant qu'exit d'API IBM MQ . Il est fourni en tant que programme de support IBM , ce qui signifie que vous avez le droit d'utiliser le service de trace et de recevoir le support dans le cadre de votre droit IBM MQ . Notez que votre autorisation d'utilisation IBM MQ autorise uniquement l'utilisation du composant d'exit de traçage IBM MQ du programme de prise en charge IBM Instana .

Les problèmes liés à l'utilisation de l'exit doivent être signalés au support IBM MQ sauf si vous utilisez l'exit de trace avec un système de surveillance IBM Instana et une autorisation IBM Instana .

L'exit peut être téléchargé ici: <https://ibm.biz/mqinstanaexit>.

## Installation et configuration du service de traçage IBM MQ Open Telemetry

Pour plus d'informations sur l'installation et la configuration de la fonction de trace IBM MQ , voir [IBM MQ Tracing](#).

Pour plus d'informations sur les plateformes prises en charge pour l'exit IBM Instana IBM MQ , voir [Plateformes prises en charge pour IBM MQ](#) sur site.

## Comportement du diagnostic d'intégrité du gestionnaire de files d'attente

Le gestionnaire de files d'attente effectue des diagnostics d'intégrité périodiques pour garantir des performances stables et fiables. Cette rubrique décrit certains des diagnostics d'intégrité effectués par le gestionnaire de files d'attente et explique comment les configurer en fonction des exigences environnementales.

Dans la plupart des environnements, la configuration par défaut convient et il n'est pas nécessaire de modifier la fréquence de ces vérifications. Même en utilisant les paramètres par défaut, il peut être utile de comprendre le comportement du gestionnaire de files d'attente lorsqu'un problème est détecté et les problèmes d'environnement qui peuvent entraîner l'échec d'une vérification. Cette rubrique explique certains de ces comportements.



**ATTENTION** : N'apportez aucune modification à la fréquence de ces vérifications, sauf si vous êtes invité à le faire par le support IBM .

Les différents composants du gestionnaire de files d'attente utilisent différents moyens pour détecter et résoudre les incohérences et cette rubrique n'est pas destinée à décrire tous les mécanismes de ce type. Par exemple, les processus IBM MQ utilisent différents mécanismes pour s'assurer que les autres processus dont ils dépendent sont toujours en cours d'exécution. Les comportements décrits sont ceux qui sont effectués périodiquement par le contrôleur d'exécution pour détecter des situations environnementales ou d'autres situations inattendues. (Le contrôleur d'exécution est le processus IBM MQ principal qui démarre et gère la plupart des autres processus de gestionnaire de files d'attente.) Etant donné qu'il s'agit de vérifications périodiques, elles sont effectuées à des intervalles spécifiques qui peuvent être modifiés dans une certaine mesure en définissant les paramètres d'optimisation appropriés.

Certaines des vérifications décrites sont effectuées par une unité d'exécution de diagnostic d'intégrité dédiée. Si un problème est détecté avec l'unité d'exécution de diagnostic d'intégrité elle-même, un message d'avertissement [AMQ5066](#) est consigné dans les journaux des erreurs du gestionnaire de files d'attente.

Les comportements décrits dans cette rubrique sont susceptibles de changer dans les éditions futures, par exemple si une valeur par défaut différente est observée comme étant plus stable sur une plateforme ou une configuration particulière.

### Contrôles de santé généraux

Le gestionnaire de files d'attente effectue diverses vérifications à intervalles réguliers. Par défaut, ces vérifications sont effectuées toutes les 10 secondes (dans certains cas, la vérification autorise deux cycles avant de signaler une erreur, ce qui se traduit par un intervalle de 20 secondes pour ces vérifications). Lors des vérifications, le gestionnaire de files d'attente s'assure que les différents processus exécutés dans le cadre du gestionnaire de files d'attente sont toujours en cours d'exécution. Pour un gestionnaire de files d'attente Native HA, il vérifie que le gestionnaire de files d'attente réplique correctement les données sur les instances de secours.

Si une vérification critique échoue à ce stade (par exemple, si le processus amqzmc0 n'est plus en cours d'exécution), le gestionnaire de files d'attente ne peut pas continuer à s'exécuter. Toutefois, la plupart

des vérifications sont effectuées pour nettoyer les ressources système qui ne sont plus nécessaires et peuvent simplement entraîner l'écriture d'un message dans les journaux d'erreurs du gestionnaire de files d'attente.

Dans la plupart des cas, il n'est pas nécessaire de modifier la fréquence de ces contrôles sanitaires généraux. La majorité des événements du gestionnaire de files d'attente ou de l'environnement d'exploitation sont détectés instantanément sans qu'il soit nécessaire que le processus de diagnostic d'intégrité général les détecte. Ce processus sert de vérification périodique de tout élément non détecté ailleurs dans le gestionnaire de files d'attente. Si nécessaire, la fréquence peut être configurée à l'aide du paramètre d'optimisation **ECHearTBeatLen**. La valeur minimale est 10000 millisecondes (10 secondes). La valeur maximale est 60000 millisecondes (60 secondes). Si la valeur maximale est 60000, cela peut entraîner un délai de deux minutes pour certaines vérifications.

## Vérifie que la progression du journal est en cours

Les vérifications du gestionnaire de files d'attente qui écrivent dans le journal sont effectuées à une fréquence raisonnable. Il ne s'agit pas de vérifier que les performances du consignateur sont optimales, mais qu'elles sont conçues pour détecter les conditions qui peuvent nécessiter une attention supplémentaire. Par exemple, si le disque sur lequel sont stockés les fichiers journaux est particulièrement lent, ou si le gestionnaire de files d'attente ne reçoit pas suffisamment de temps UC dans un environnement conteneurisé pour effectuer tout son travail.

Si cette vérification échoue, l'action effectuée par le gestionnaire de files d'attente dépend du type de gestionnaire de files d'attente utilisé:

- Sur un gestionnaire de files d'attente sans haute disponibilité:
  - Un FDC `xecl_W_PERFORMANCE_BOTTLENECK` est écrit. Cela peut être utilisé pour indiquer qu'une partie du système peut nécessiter une attention supplémentaire. Le gestionnaire de files d'attente reste en cours d'exécution. Si des FDC `xecl_W_PERFORMANCE_BOTTLENECK` sont visibles dans le répertoire des erreurs, il peut être nécessaire de travailler avec votre équipe de stockage ou de plateforme pour déterminer si les ressources système sous-jacentes sont suffisantes pour que IBM MQ s'exécute. Si IBM MQ est exécuté dans des conteneurs sur des noeuds surchargés, il se peut que IBM MQ ne reçoive pas suffisamment de temps UC planifié pour exécuter l'ensemble de sa charge de travail de messagerie.
  - Depuis la IBM MQ 9.3.0, un message d'avertissement `AMQ5068W` est consigné dans les journaux des erreurs du gestionnaire de files d'attente et aucun FDC `xecl_W_PERFORMANCE_BOTTLENECK` n'est consigné. Si des messages `AMQ5068W` sont affichés dans les journaux, il peut être nécessaire de collaborer avec votre équipe de stockage ou de plateforme pour déterminer si les ressources système sous-jacentes sont suffisantes pour que IBM MQ s'exécute. Si IBM MQ est exécuté dans des conteneurs sur des noeuds surchargés, il se peut que IBM MQ ne reçoive pas suffisamment de temps UC planifié pour exécuter l'ensemble de sa charge de travail de messagerie. Si cinq messages d'avertissement `AMQ5068W` sont écrits successivement, un FDC `xecl_W_PERFORMANCE_BOTTLENECK` est écrit.
- Sur un gestionnaire de files d'attente multi-instance:
  - Si le diagnostic d'intégrité de la progression du journal échoue, l'instance principale se termine. Si une instance de secours est disponible, elle démarre et devient l'instance principale.
  - Depuis la IBM MQ 9.3.0, l'instance principale vérifie si une instance de secours est disponible avant de s'arrêter. Si un gestionnaire de files d'attente de secours est disponible pour le basculement vers l'instance principale, il s'arrête. En outre, un message d'avertissement `AMQ5068W` est consigné dans les journaux des erreurs du gestionnaire de files d'attente.
- Sur un gestionnaire de files d'attente Native HA, cette vérification se comporte de la même manière qu'un gestionnaire de files d'attente sans haute disponibilité.
- Sur un gestionnaire de files d'attente de données répliquées (RDQM), cette vérification se comporte de la même manière qu'un gestionnaire de files d'attente sans haute disponibilité.

Il est possible qu'un problème lié à la progression du journal IBM MQ soit causé par un problème de performances dans le gestionnaire de files d'attente lui-même.

Par défaut, cette vérification est effectuée toutes les 60 secondes, bien que le gestionnaire de files d'attente attende deux cycles de vérification avant d'effectuer une action. Cela signifie qu'avec les paramètres par défaut, deux minutes doivent s'écouler avant que le gestionnaire de files d'attente écrive un message d'erreur (ou qu'il bascule dans le cas d'un gestionnaire de files d'attente à haute disponibilité).

Dans la plupart des cas, le comportement par défaut convient même lorsque le système de fichiers est lent ou que le gestionnaire de files d'attente se voit allouer une petite quantité de temps UC car d'autres vérifications telles que le verrouillage de fichier (voir «[Vérifie que les verrous de fichier sont toujours maintenus](#)», à la page 8) et le fonctionnement du système de fichiers de base entraînent la reprise en ligne d'une instance principale avant cette vérification. Si nécessaire, la fréquence de cette vérification peut être configurée à l'aide du paramètre d'optimisation **LivenessHeartBeatLen**. La valeur maximale sur laquelle il peut être configuré est de 600 secondes (10 minutes). La valeur minimale de 0 a pour effet de désactiver complètement la vérification. Pour un gestionnaire de files d'attente sans haute disponibilité, le seul effet de la vérification est un message d'avertissement supplémentaire dans les journaux des erreurs du gestionnaire de files d'attente. Pour un gestionnaire de files d'attente multi-instance, vous pouvez configurer **LivenessHeartBeatLen** pour qu'une instance principale du gestionnaire de files d'attente soit basculée plus rapidement (en réduisant la valeur) ou lentement (en augmentant la valeur). L'augmentation de la valeur pour réduire la fréquence de la vérification de la progression du journal peut être utile si votre environnement connaît parfois des E-S de système de fichiers très lentes, mais que vous préférez que l'instance principale du gestionnaire de files d'attente reste en cours d'exécution. Cela peut être utile si vous disposez d'applications qui ne sont pas conçues pour se reconnecter automatiquement à l'instance de secours et qui nécessitent une intervention manuelle pour les redémarrer.

**Remarque :** Si le **ECHearBeatLen** a été augmenté, cela affecte la temporisation des vérifications **LivenessHeartBeatLen**. Les vérifications de la progression du journal sont effectuées lorsque les vérifications de santé générales sont effectuées, de sorte que la réduction de la fréquence des vérifications de santé générales (**ECHearBeatLen**) peut entraîner des vérifications de la progression du journal jusqu'à 30 secondes après la **LivenessHeartBeatLen** configurée.

## Performances générales du système de fichiers de journalisation

► V 9.4.0

Depuis la IBM MQ 9.4.0, un message d'avertissement [AMQ6729W](#) est émis dans le journal des erreurs du gestionnaire de files d'attente si les opérations générales de lecture / écriture sur ce stockage semblent prendre plus de temps que prévu. Vous pouvez utiliser les variables d'environnement **AMQ\_IODELAY** pour optimiser les diagnostics et les délais afin d'aider à diagnostiquer les problèmes de performances de stockage ou d'augmenter la tolérance à ces retards. Pour plus d'informations, voir [AMQ\\_IODELAY](#), [AMQ\\_IODELAY\\_INMS](#) et [AMQ\\_IODELAY\\_FFST](#).

## Vérifie que les verrous de fichier sont toujours maintenus

Pour un gestionnaire de files d'attente multi-instance, le contrôleur d'exécution vérifie régulièrement qu'il détient toujours le verrou exclusif sur le fichier multi-instance principal. Dans de nombreux cas, si le verrou est perdu en raison d'un problème avec le serveur NFS, l'instance principale bascule presque immédiatement (avant cette vérification). Des contrôles de verrouillage de fichier périodiques supplémentaires sont effectués pour s'assurer que le gestionnaire de files d'attente principal est défaillant en cas de problème de système de fichiers inhabituel.

Par défaut, ces vérifications de verrouillage de fichier sont effectuées toutes les 20 secondes. Si nécessaire, vous pouvez modifier cette valeur en définissant le paramètre d'optimisation **FileLockHeartBeatLen**. La valeur par défaut du paramètre d'optimisation est de 10 secondes (le gestionnaire de files d'attente autorise deux cycles de vérification avant d'effectuer une action entraînant le comportement par défaut de la vérification toutes les 20 secondes). La valeur minimale du paramètre d'optimisation est de 10 secondes, la valeur maximale est de 600 secondes (10 minutes).

**Remarque :** Si le **ECHearBeatLen** a été augmenté, cela affecte la temporisation des vérifications **FileLockHeartBeatLen**. Les vérifications de verrouillage de fichier sont effectuées lorsque les



vérifications de santé générale sont effectuées, de sorte que la réduction de la fréquence des vérifications de santé générale (**ECHeartBeatLen**) peut entraîner des vérifications de verrouillage de fichier jusqu'à 30 secondes après la **FileLockHeartBeatLen** configurée.

## Vérifications de la santé de l'application utilisateur

Le gestionnaire de files d'attente vérifie régulièrement que toutes les applications liées localement qui ne sont plus en cours d'exécution ont effectué un appel MQI MQDISC avant de s'arrêter. Ces vérifications sont effectuées en même temps que les vérifications de santé générales décrites dans «Contrôles de santé généraux», à la page 6. L'intervalle par défaut pour ces vérifications est donc de 10000 millisecondes (10 secondes) et la modification de la valeur du paramètre d'optimisation **ECHeartBeatLen** modifie la fréquence à laquelle elles sont effectuées. Cette vérification vise principalement à s'assurer que les ressources associées à une application connectée sont libérées. Elle n'entraîne pas l'arrêt d'un gestionnaire de files d'attente à haute disponibilité ou d'un gestionnaire de files d'attente non à haute disponibilité ou le basculement vers une autre instance.

Les applications client IBM MQ qui se sont arrêtées sans émettre d'appel MQI MQDISC sont détectées séparément par le processus agent et toutes les ressources associées à la connexion sont libérées.

### Concepts associés

[Configurations à haute disponibilité](#)

## Moniteur d'événements

La surveillance des événements consiste à détecter les occurrences d' *événements d'instrumentation* dans un réseau de gestionnaires de files d'attente. Un événement d'instrumentation est une combinaison logique d'événements détectés par un gestionnaire de files d'attente ou une instance de canal. Un tel événement entraîne le gestionnaire de files d'attente ou l'instance de canal à insérer un message spécial, appelé *message d'événement*, dans une file d'attente d'événements.

Les événements d'instrumentation IBM MQ fournissent des informations sur les erreurs, les avertissements et d'autres occurrences importantes dans un gestionnaire de files d'attente. Utilisez ces événements pour surveiller le fonctionnement des gestionnaires de files d'attente dans votre réseau de gestionnaires de files d'attente afin d'atteindre les objectifs suivants:

- Détectez les problèmes dans votre réseau de gestionnaires de files d'attente.
- Aide à déterminer les causes des problèmes dans votre réseau de gestionnaires de files d'attente.
- Générez une trace d'audit.
- Réagir aux changements d'état du gestionnaire de files d'attente

### Référence associée

«Types d'événements», à la page 12

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

[Référence de message d'événement](#)

[Format de message d'événement](#)

## Publication de vos messages d'événement IBM MQ

Comment préparer IBM MQ à la publication de messages d'événement.

## Pourquoi et quand exécuter cette tâche

Les messages d'événement sont écrits dans des files d'attente nommées spécialement `SYSTEM.ADMIN.<feature name>.EVENT`.

L'important à noter sur ces files d'attente d'événements est que c'est le nom qui compte. Par défaut, sur un gestionnaire de files d'attente, toutes les files d'attente d'événements sont définies en tant que files d'attente locales. Toutefois, vous pouvez supprimer ces files d'attente et les redéfinir, peut-être en

tant que file d'attente éloignée, de sorte que tous les événements soient placés dans un gestionnaire de files d'attente de traitement d'événements dédié. Vous pouvez également utiliser une file d'attente alias pointant vers un objet de rubrique.

Dans les deux cas, toute technique de redirection nécessite que vos applications qui lisent les files d'attente d'événements n'aient pas codé en dur le nom de la file d'attente à lire. Par conséquent, vous devez pouvoir configurer la file d'attente à partir de laquelle les applications sont lues.

Les commandes suivantes montrent comment vous pouvez redéfinir vos files d'attente d'événements de sorte que les messages d'événement soient publiés, en utilisant les hypothèses suivantes. Vous avez:

- Non démarré à l'aide d'événements, ou
- Tous les messages des files d'attente d'événements existantes ont été supprimés et les files d'attente locales ont été supprimées avant ces étapes.

Ces étapes montrent uniquement les files d'attente d'événements QMGR et CHANNEL en cours de redéfinition, mais elles peuvent être étendues pour tous les événements.

**Remarque :** La chaîne de rubrique est conçue pour qu'une application puisse être abonnée à tous les événements à l'aide d'un caractère générique ou à des événements spécifiques, selon les besoins.

## Procédure

Exécutez les commandes suivantes :

```
DEFINE TOPIC (ADMIN.QMGR.EVENT) TOPICSTR('Events/QMgr')
DEFINE TOPIC (ADMIN.CHANNEL.EVENT) TOPICSTR('Events/Channel')

DEFINE QALIAS (SYSTEM.ADMIN.QMGR.EVENT) TARGTYPE(TOPIC) TARGET (ADMIN.QMGR.EVENT)
DEFINE QALIAS (SYSTEM.ADMIN.CHANNEL.EVENT) TARGTYPE(TOPIC) TARGET (ADMIN.CHANNEL.EVENT)

DEFINE QLOCAL (ADMIN.EVENT)
DEFINE QLOCAL (ADMIN.QMGR.EVENT)

DEFINE SUB (EVENTS.ALL) TOPICSTR('Events+') PSPROP(NONE)
DESTCLAS (PROVIDED) DEST (ADMIN.EVENT)
DEFINE SUB (EVENTS.QMGR) TOPICSTR('Events/QMgr') PSPROP(NONE)
DESTCLAS (PROVIDED) DEST (ADMIN.QMGR.EVENT)
```

En supposant que votre application de lecture d'événement est capable de lire des messages d'événement à partir de n'importe quelle file d'attente, cette application peut être reconfigurée pour lire l'une des files d'attente définies ci-dessus selon les besoins.

La configuration `PSPROP(NONE)` des commandes `DEFINE SUB` permet de s'assurer qu'aucune des propriétés de message ajoutées par le moteur de publication / abonnement, par exemple `MQTopicString`, n'est ajoutée au message d'événement, ce qui garantit que les applications existantes peuvent continuer à fonctionner sans modification.

De plus, les applications peuvent également s'abonner directement à l'aide de l'appel `MQSUB` pour recevoir les informations, de manière alternative à l'utilisation de la commande administrative `DEFINE SUB`.

Désormais, plusieurs applications peuvent consommer les informations émises dans les événements par le gestionnaire de files d'attente.

## Événements d'instrumentation

Un événement d'instrumentation est une combinaison logique de conditions qu'un gestionnaire de files d'attente ou une instance de canal détecte et insère un message spécial, appelé *message d'événement*, dans une file d'attente d'événements.

Les événements d'instrumentation IBM MQ fournissent des informations sur les erreurs, les avertissements et d'autres occurrences importantes dans un gestionnaire de files d'attente. Vous pouvez utiliser ces événements pour surveiller le fonctionnement des gestionnaires de files d'attente (avec d'autres méthodes telles que Tivoli NetView for z/OS).

La [Figure 1](#), à la [page 11](#) illustre le concept d'événements d'instrumentation.

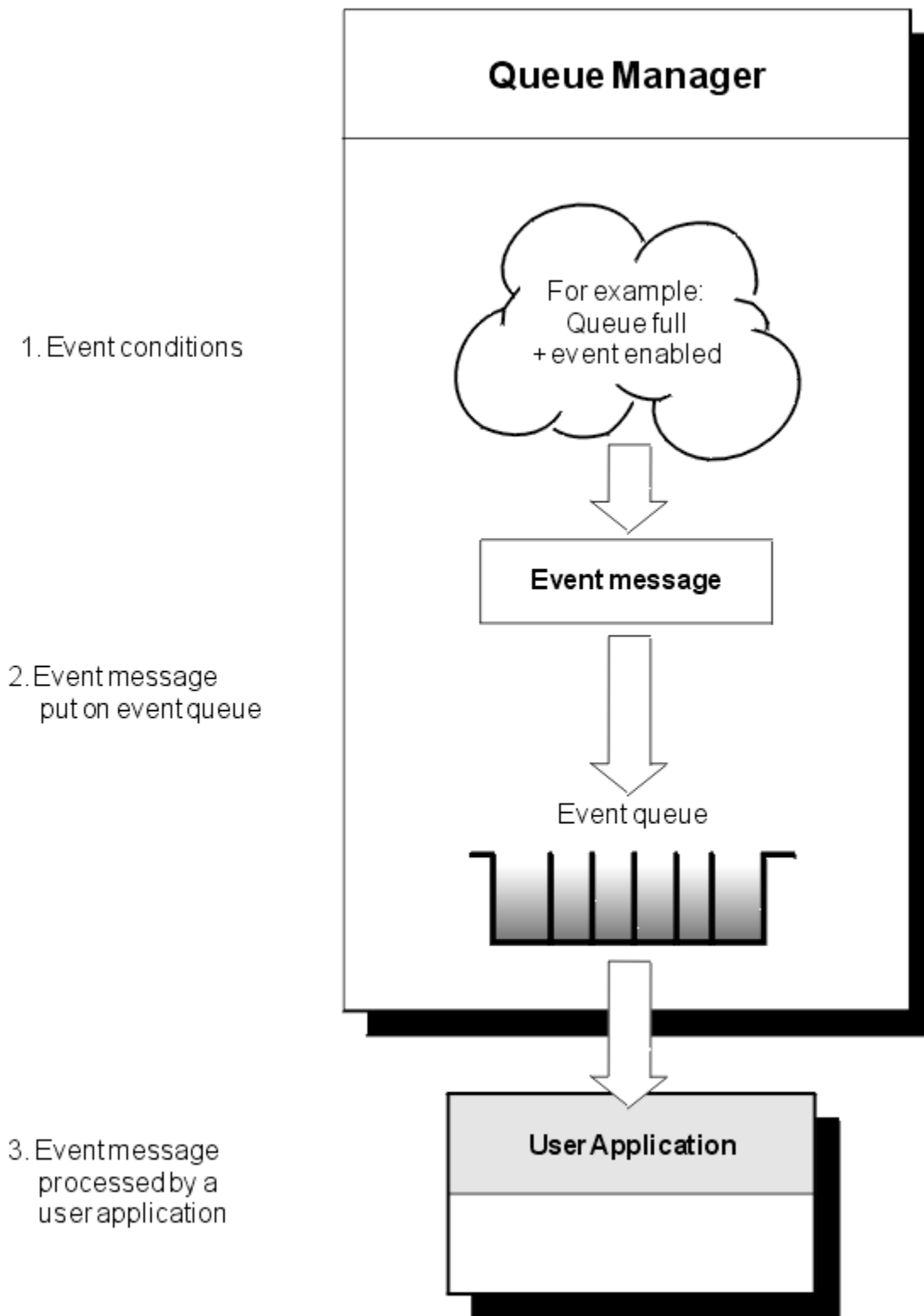


Figure 1. Description des événements d'instrumentation

### Applications de surveillance des événements

Les applications qui utilisent des événements pour surveiller les gestionnaires de files d'attente doivent inclure les dispositions suivantes:

1. Configurez des canaux entre les gestionnaires de files d'attente de votre réseau.

2. Implémentez les conversions de données requises. Les règles normales de conversion des données s'appliquent. Par exemple, si vous surveillez des événements sur un gestionnaire de files d'attente système UNIX à partir d'un gestionnaire de files d'attente z/OS , veillez à convertir EBCDIC en ASCII.

## Notification d'événements via des files d'attente d'événements

Lorsqu'un événement se produit, le gestionnaire de files d'attente place un message d'événement dans la file d'attente d'événements appropriée, si elle est définie. Le message d'événement contient des informations sur l'événement que vous pouvez extraire en écrivant un programme d'application MQI approprié qui effectue les étapes suivantes:

- Extrait le message de la file d'attente.
- Traitez le message pour extraire les données d'événement.

Les informations connexes décrivent le format des messages d'événement.

## Conditions à l'origine des événements

La liste suivante fournit des exemples de conditions pouvant provoquer des événements d'instrumentation:

- Une limite de seuil pour le nombre de messages dans une file d'attente est atteinte.
- Une instance de canal est démarrée ou arrêtée.
- Un gestionnaire de files d'attente devient actif ou doit être arrêté.
- Une application tente d'ouvrir une file d'attente en spécifiant un ID utilisateur qui n'est pas autorisé sur les systèmes IBM MQ for IBM i, AIX, Linux®, and Windows .
- Les objets sont créés, supprimés, modifiés ou actualisés.
- Une commande MQSC ou PCF s'exécute correctement.
- Un gestionnaire de files d'attente commence à écrire dans un nouveau domaine réservé aux journaux.
- Insertion d'un message dans la file d'attente de rebut, si les conditions d'événement sont remplies.

### Concepts associés

[«Événements Performances», à la page 25](#)

Les événements de performances sont liés à des conditions qui peuvent affecter les performances des applications qui utilisent une file d'attente spécifiée. La portée des événements de performances est la file d'attente. Les appels **MQPUT** et **MQGET** sur une file d'attente n'affectent pas la génération d'événements de performances sur une autre file d'attente.

[«Exemple de programme de surveillance des événements d'instrumentation sur Multiplatforms», à la page 59](#)

**amqsevt** formate les événements d'instrumentation qu'un gestionnaire de files d'attente peut créer et est fourni avec IBM MQ for Multiplatforms. Le programme lit les messages des files d'attente d'événements et les convertit en chaînes lisibles.

## Types d'événements

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

Les types des événements d'instrumentation IBM MQ sont les suivants:

- Événements du gestionnaire de files d'attente
- Événements de canal et de pont
- Événements Performances
- Événements de configuration
- Événements Commande
- Événements de journalisation

- Événements locaux

Pour chaque gestionnaire de files d'attente, chaque catégorie d'événement possède sa propre file d'attente d'événements. Tous les événements de cette catégorie entraînent l'insertion d'un message d'événement dans la même file d'attente.

**Cette file d'attente d'événements:**

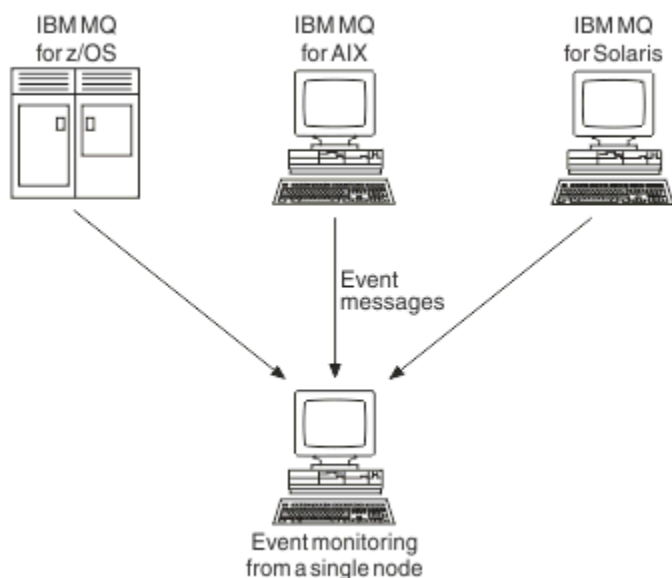
SYSTEM.ADMIN.QMGR.EVENT  
 SYSTEM.ADMIN.CHANNEL.EVENT  
 SYSTEM.ADMIN.PERFM.EVENT  
 SYSTEM.ADMIN.CONFIG.EVENT  
 SYSTEM.ADMIN.COMMAND.EVENT  
 SYSTEM.ADMIN.LOGGER.EVENT  
 SYSTEM.ADMIN.PUBSUB.EVENT

**Contient des messages provenant de:**

Événements du gestionnaire de files d'attente  
 Événements Canal  
 Événements Performances  
 Événements de configuration  
 Événements Commande  
 Événements de journalisation  
 Obtient les événements liés à la publication /  
 abonnement. Utilisé uniquement avec la  
 multidiffusion. Pour plus d'informations, voir  
[Surveillance des applications multidiffusion.](#)

En incorporant des événements d'instrumentation dans votre propre application de gestion de système, vous pouvez surveiller les activités sur de nombreux gestionnaires de files d'attente, sur de nombreux noeuds et pour plusieurs applications IBM MQ . En particulier, vous pouvez surveiller tous les noeuds de votre système à partir d'un seul noeud (pour les noeuds qui prennent en charge les événements IBM MQ ), comme illustré dans la [Figure 2](#), à la [page 13](#).

Les événements d'instrumentation peuvent être signalés via un mécanisme de génération de rapports écrit par l'utilisateur à une application d'administration qui peut présenter les événements à un opérateur.



*Figure 2. Surveillance des gestionnaires de files d'attente sur différentes plateformes, sur un seul noeud*

Les événements d'instrumentation permettent également aux applications agissant en tant qu'agents pour d'autres réseaux d'administration, par exemple Tivoli NetView for z/OS, de surveiller les rapports et de créer les alertes appropriées.

*Événements du gestionnaire de files d'attente*

Les événements de gestionnaire de files d'attente sont liés à l'utilisation des ressources dans les gestionnaires de files d'attente. Par exemple, un événement de gestionnaire de files d'attente est généré si une application tente d'insérer un message dans une file d'attente qui n'existe pas.

Les exemples suivants illustrent des conditions pouvant entraîner un événement de gestionnaire de files d'attente:

- Une application émet un appel MQI qui échoue. Le code anomalie de l'appel est identique à celui du message d'événement.

Une condition similaire peut se produire lors de l'opération interne d'un gestionnaire de files d'attente ; par exemple, lors de la génération d'un message de rapport. Le code anomalie d'un message d'événement peut correspondre à un code anomalie MQI, même s'il n'est associé à aucune application. Ne partez pas du principe que, comme un code anomalie de message d'événement ressemble à un code anomalie MQI, l'événement a nécessairement été provoqué par un appel MQI ayant échoué à partir d'une application.

- Une commande est émise vers un gestionnaire de files d'attente et le traitement de cette commande entraîne un événement. Exemple :
  - Un gestionnaire de files d'attente est arrêté ou démarré.
  - Une commande est émise lorsque l'ID utilisateur associé n'est pas autorisé pour cette commande.

IBM MQ insère des messages pour les événements du gestionnaire de files d'attente dans SYSTEM.ADMIN.QMGR.EVENT et prend en charge les types d'événement de gestionnaire de files d'attente suivants:

#### **ALW** Droits d'accès (sous AIX, Linux, and Windows uniquement)

Les événements de droits signalent une autorisation, telle qu'une application qui tente d'ouvrir une file d'attente pour laquelle elle ne dispose pas des droits requis, ou une commande émise à partir d'un ID utilisateur qui ne dispose pas des droits requis. Le message d'événement de droits d'accès peut contenir les données d'événement suivantes:

- [Non autorisé \(type 1\)](#)
- [Non autorisé \(type 2\)](#)
- [Non autorisé \(type 3\)](#)
- [Non autorisé \(type 4\)](#)
- [Non autorisé \(type 5\)](#)
- [Non autorisé \(type 6\)](#)

Tous les événements de droits d'accès sont valides sur AIX, Linux, and Windows uniquement.

#### **Bloquer**

Les événements d'interdiction indiquent qu'une opération MQPUT ou MQGET a été tentée sur une file d'attente dans laquelle la file d'attente est interdite pour les insertions ou les extractions, ou sur une rubrique dans laquelle la rubrique est interdite pour les publications. Le message d'événement d'interdiction peut contenir les données d'événement suivantes:

- [Extraction interdite](#)
- [Insertion interdite](#)

#### **Locale**

Lorsqu'une application ou le gestionnaire de files d'attente n'a pas pu accéder à une file d'attente locale ou à un autre objet local, par exemple parce que l'objet n'a pas été défini, le gestionnaire de files d'attente peut générer un message d'événement local. Le message d'événement local peut contenir les données d'événement suivantes:

- [Erreur de type de file d'attente de base alias](#)
- [File d'attente de base inconnue](#)
- [Nom d'objet inconnu](#)

#### **Eloigné**


Lorsqu'une application ou le gestionnaire de files d'attente ne peut pas accéder à une file d'attente éloignée d'un autre gestionnaire de files d'attente, par exemple lorsque la file d'attente de transmission n'est pas correctement définie, le gestionnaire de files d'attente peut générer

un message d'événement éloigné. Le message d'événement distant peut contenir les données d'événement suivantes:

- [Erreur de type de file d'attente de transmission par défaut](#)
- [Erreur d'utilisation de la file d'attente de transmission par défaut](#)
- [Erreur de type de file d'attente](#)
- [Erreur de nom de file d'attente distante](#)
- [Erreur de type de file d'attente de transmission](#)
- [Erreur d'utilisation de file d'attente de transmission](#)
- [File d'attente de transmission par défaut inconnue](#)
- [Gestionnaire de files d'attente distant inconnu](#)
- [File d'attente de transmission inconnue](#)

### Démarrage et arrêt

Les événements de démarrage et d'arrêt indiquent qu'un gestionnaire de files d'attente a été démarré ou qu'il a été demandé d'arrêter ou de mettre au repos.

 z/OS ne prend en charge que les événements de début.

Les événements d'arrêt ne sont enregistrés que si la persistance de message par défaut de SYSTEM.ADMIN.QMGR.EVENT est définie comme persistante. Le message d'événement de démarrage et d'arrêt peut contenir les données d'événement suivantes:

- [Gestionnaire de files d'attente actif](#)
- [Gestionnaire de files d'attente non actif](#)

Pour chaque type d'événement de cette liste, vous pouvez définir un attribut de gestionnaire de files d'attente pour activer ou désactiver le type d'événement.

### *Evénements de canal et de pont*

Les canaux signalent ces événements à la suite de conditions détectées lors de leur fonctionnement. Par exemple, lorsqu'une instance de canal est arrêtée.

Les événements de canal sont générés dans les cas suivants:

- Lorsqu'une commande démarre ou arrête un canal.
- Lorsqu'une instance de canal démarre ou s'arrête.
- Lorsqu'un canal reçoit un avertissement d'erreur de conversion lors de l'obtention d'un message.
- Lorsqu'une tentative de création automatique d'un canal est effectuée, l'événement est généré si la tentative aboutit ou échoue.

**Remarque :** Les connexions client ne provoquent pas d'événements Canal démarré ou Canal arrêté.

Lorsqu'une commande est utilisée pour démarrer un canal, un événement est généré. Un autre événement est généré lorsque l'instance de canal démarre. Toutefois, le démarrage d'un canal par un programme d'écoute, la commande **runmqchl** ou un message de déclenchement de gestionnaire de files d'attente ne génère pas d'événement. Dans ces cas, un événement est généré uniquement lorsque l'instance de canal démarre.

Une commande de démarrage ou d'arrêt de canal réussie génère au moins deux événements. Ces événements sont générés pour les deux gestionnaires de files d'attente connectés par le canal (à condition qu'ils prennent en charge les événements).

Si un événement de canal est placé dans une file d'attente d'événements, une condition d'erreur entraîne la création d'un événement par le gestionnaire de files d'attente.

Les messages d'événement pour les événements de canal et de pont sont placés dans SYSTEM.ADMIN.CHANNEL.EVENT EVENT.

Les messages d'événement de canal peuvent contenir les données d'événement suivantes:

- [Canal activé](#)
- [Erreur de définition automatique de canal](#)
- [Définition automatique de canal OK](#)
- [Erreur de conversion de canal](#)
- [Canal non activé](#)
- [Canal démarré](#)
- [Canal arrêté](#)
- [Canal arrêté par l'utilisateur](#)
- [Canal bloqué](#)

## Événements de pont IMS ( z/OS uniquement)



Ces événements sont signalés lorsqu'un pont IMS démarre ou s'arrête.

Les messages d'événement de pont IMS peuvent contenir les données d'événement suivantes:

- [Pont démarré](#)
- [Pont arrêté](#)

## Événements SSL

Le seul événement TLS est l'événement Erreur SSL du canal. Cet événement est signalé lorsqu'un canal utilisant TLS ne parvient pas à établir une connexion TLS.

Les messages d'événement SSL peuvent contenir les données d'événement suivantes:

- [Erreur SSL de canal](#)
- [Avertissement SSL du canal](#)

### *Événements Performances*

Les événements de performances sont des notifications indiquant qu'une ressource a atteint une condition de seuil. Par exemple, une limite de longueur de file d'attente a été atteinte.

Les événements de performances sont liés à des conditions qui peuvent affecter les performances des applications qui utilisent une file d'attente spécifiée. Ils ne sont pas générés pour les files d'attente d'événements elles-mêmes.

Le type d'événement est renvoyé dans la zone d'identificateur de commande des données de message.

Si un gestionnaire de files d'attente tente d'insérer un événement de gestionnaire de files d'attente ou un message d'événement de performances dans une file d'attente d'événements et qu'une erreur qui crée généralement un événement est détectée, un autre événement n'est pas créé et aucune action n'est effectuée.

Les appels MQGET et MQPUT au sein d'une unité de travail peuvent générer des événements de performances, que l'unité de travail soit validée ou annulée.

Les messages d'événement pour les événements de performances sont placés dans SYSTEM.ADMIN.PERFM.EVENT EVENT.

Il existe deux types d'événement de performance:

### **Événements de longueur de file d'attente**

Les événements de longueur de file d'attente sont liés au nombre de messages dans une file d'attente, c'est-à-dire à la saturation ou à la mise à vide de la file d'attente. Ces événements sont pris en charge pour les files d'attente partagées. Les messages d'événement de longueur de file d'attente peuvent contenir les données d'événement suivantes:


- [Longueur élevée de file d'attente](#)



- Longueur faible de file d'attente
- File d'attente saturée

### **Événements Intervalle de maintenance de file d'attente**

Les événements d'intervalle de service de file d'attente indiquent si les messages sont traités dans un intervalle de temps spécifié par l'utilisateur. Ces événements ne sont pas pris en charge pour les files d'attente partagées.

 IBM MQ for z/OS prend en charge les événements de longueur de file d'attente pour les files d'attente QSGDISP (SHARED), mais pas les événements d'intervalle de service. Les événements de gestionnaire de files d'attente et de canal ne sont pas affectés par les files d'attente partagées. Les messages d'événement du service de file d'attente peuvent contenir les données d'événement suivantes:

- Intervalle de service de file d'attente important
- Intervalle de service de file d'attente OK

### *Événements de configuration*

Les événements de configuration sont générés lorsqu'un événement de configuration est demandé explicitement ou automatiquement lorsqu'un objet est créé, modifié ou supprimé.

Un message d'événement de configuration contient des informations sur les attributs d'un objet. Par exemple, un message d'événement de configuration est généré si un objet liste de noms est créé et contient des informations sur les attributs de l'objet liste de noms.

Les messages d'événement pour les événements de configuration sont placés dans SYSTEM.ADMIN.CONFIG.EVENT EVENT.

Les événements de configuration fonctionnent de la manière suivante:

- Une trace d'audit des modifications apportées à la configuration du gestionnaire de files d'attente est conservée au moyen de messages d'événement, qui sont écrits dans SYSTEM.ADMIN.CONFIG.EVENT EVENT. Vous pouvez activer l'écriture de ces événements à l'aide du paramètre **CONFIGEV** de la commande ALTER QMGR.
- Ces événements sont générés lorsqu'une commande DEFINE, ALTER ou DELETE agit sur un objet ou qu'un appel MQSET est utilisé.
- Vous pouvez créer une image de ligne de base de la configuration de gestionnaire de files d'attente en cours à l'aide de la commande REFRESH QMGR TYPE (CONFIGEV), qui crée un message d'événement pour chaque objet du gestionnaire de files d'attente. Etant donné que cette opération peut prendre du temps, si vous avez de nombreux objets, vous pouvez fractionner la tâche en plus petits ensembles d'objets à l'aide des qualificatifs NAME et OBJECT de la commande.
- Le message d'événement enregistre l'une des quatre raisons possibles:

- MQRC\_CONFIG\_CHANGE\_OBJECT
- MQRC\_CONFIG\_CREATE\_OBJECT
- MQRC\_CONFIG\_DELETE\_OBJECT
- MQRC\_CONFIG\_REFRESH\_OBJECT

MQRC\_CONFIG\_CHANGE\_OBJECT, MQRC\_CONFIG\_CREATE\_OBJECT ou MQRC\_CONFIG\_DELETE\_OBJECT se produisent pour les commandes MQSC ou PCF que vous pouvez émettre sur un objet.

MQRC\_CONFIG\_REFRESH\_OBJECT se produit pour les messages d'événement écrits lors de la création de l'image de ligne de base.

Il existe quatre types d'événement de configuration:

### **Créer des événements d'objet**

Les événements de création d'objet sont générés lorsqu'un objet est créé. Le message d'événement contient les données d'événement suivantes: Create object.

### Événements de modification d'objet

Les événements de modification d'objet sont générés lorsqu'un objet est modifié. Le message d'événement contient les données d'événement suivantes: [Changer l'objet](#).

### Événements de suppression d'objet

Les événements de suppression d'objet sont générés lorsqu'un objet est supprimé. Le message d'événement contient les données d'événement suivantes: [Supprimer l'objet](#).

### Actualiser les événements d'objet

Les événements d'actualisation d'objet sont générés par une demande explicite d'actualisation. Le message d'événement contient les données d'événement suivantes: [Actualiser l'objet](#).

#### *Événements Commande*

Les événements de commande sont signalés lorsqu'une commande MQSC ou PCF s'exécute correctement.

Un message d'événement de commande contient des informations sur l'origine, le contexte et le contenu d'une commande. Par exemple, un message d'événement de commande est généré avec ces informations si la commande MQSC, ALTER QLOCAL, s'exécute correctement.

Les messages d'événement des événements de commande sont placés dans SYSTEM.ADMIN.COMMAND.EVENT EVENT.

Les événements de commande contiennent les données d'événement suivantes: [Commande](#).

#### **Multi** *Événements de journalisation*

Les événements du consignateur sont signalés lorsqu'un gestionnaire de files d'attente qui utilise la journalisation linéaire commence à écrire des enregistrements de journal dans une nouvelle extension de journal **IBM i** ou, sous IBM i, dans un nouveau récepteur de journal. **z/OS** Les événements du consignateur ne sont pas disponibles avec IBM MQ for z/OS.

Un message d'événement de consignateur contient des informations indiquant les extensions de journal requises par le gestionnaire de files d'attente pour redémarrer le gestionnaire de files d'attente ou pour la reprise sur incident lié au support.

Les messages d'événement pour les événements du consignateur sont placés dans SYSTEM.ADMIN.LOGGER.EVENT EVENT.

Le message d'événement du consignateur contient les données d'événement suivantes: [Consignateur](#).

#### *Récapitulatif des données de message d'événement*

Utilisez ce récapitulatif pour obtenir des informations sur les données d'événement que chaque type de message d'événement peut contenir.

Type d'événement	Voir ces rubriques
Événements de droit d'accès	<a href="#">Non autorisé (type 1)</a>
	<a href="#">Non autorisé (type 2)</a>
	<a href="#">Non autorisé (type 3)</a>
	<a href="#">Non autorisé (type 4)</a>
	<a href="#">Non autorisé (type 5)</a>
	<a href="#">Non autorisé (type 6)</a>

Type d'événement	Voir ces rubriques
Evénements Canal	<a href="#">Canal activé</a>
	<a href="#">Erreur de définition automatique de canal</a>
	<a href="#">Définition automatique de canal OK</a>
	<a href="#">Canal bloqué</a>
	<a href="#">Erreur de conversion de canal</a>
	<a href="#">Canal non activé</a>
	<a href="#">Canal démarré</a>
	<a href="#">Canal arrêté</a>
	<a href="#">Canal arrêté par l'utilisateur</a>
Evénements Commande	<a href="#">Commande</a>
Evénements de configuration	<a href="#">Créer un objet</a>
	<a href="#">Modifier un objet</a>
	<a href="#">Supprimer l'objet</a>
	<a href="#">Régénérer l'objet</a>
Evénements de pont IMS	<a href="#">Pont démarré</a>
	<a href="#">Pont arrêté</a>
Evénements Interdiction	<a href="#">Extraction interdite</a>
	<a href="#">Insertion interdite</a>
Evénements locaux	<a href="#">Erreur de type de file d'attente de base alias</a>
	<a href="#">File d'attente de base inconnue</a>
	<a href="#">Nom d'objet inconnu</a>
Evénements de journalisation	<a href="#">Consignateur</a>
Evénements Performances	<a href="#">Longueur élevée de file d'attente</a>
	<a href="#">Longueur faible de file d'attente</a>
	<a href="#">File d'attente saturée</a>
	<a href="#">Intervalle de service de file d'attente important</a>
	<a href="#">Intervalle de service de file d'attente OK</a>

Type d'événement	Voir ces rubriques
Événements éloignés	<a href="#">Erreur de type de file d'attente de transmission par défaut</a>
	<a href="#">Erreur d'utilisation de la file d'attente de transmission par défaut</a>
	<a href="#">Erreur de type de file d'attente</a>
	<a href="#">Erreur de nom de file d'attente distante</a>
	<a href="#">Erreur de type de file d'attente de transmission</a>
	<a href="#">Erreur d'utilisation de file d'attente de transmission</a>
	<a href="#">File d'attente de transmission par défaut inconnue</a>
	<a href="#">Gestionnaire de files d'attente distant inconnu</a>
	<a href="#">File d'attente de transmission inconnue</a>
Événements SSL	<a href="#">Erreur SSL de canal</a>
Événements d'arrêt et de démarrage	<a href="#">Gestionnaire de files d'attente actif</a>
	<a href="#">Gestionnaire de files d'attente non actif</a>


### Contrôle des événements

Vous activez et désactivez des événements en spécifiant les valeurs appropriées pour le gestionnaire de files d'attente, les attributs de file d'attente ou les deux, en fonction du type d'événement.

Vous devez activer chaque événement d'instrumentation que vous souhaitez générer. Par exemple, les conditions à l'origine d'un événement File d'attente saturée sont les suivantes:

- Les événements de file d'attente saturée sont activés pour une file d'attente spécifiée, et
- Une application émet une demande MQPUT pour placer un message dans cette file d'attente, mais la demande échoue car la file d'attente est saturée.

Activez et désactivez les événements à l'aide de l'une des techniques suivantes:

- Commandes de script IBM MQ (MQSC).
- Commandes PCF IBM MQ correspondantes.
-  Les panneaux d'opérations et de contrôle des gestionnaires de files d'attente sous z/OS.
- IBM MQ Explorer.

**Remarque :** Vous pouvez définir des attributs liés aux événements pour les files d'attente et les gestionnaires de files d'attente uniquement par commande. L'appel MQI MQSET ne prend pas en charge les attributs liés aux événements.

### Concepts associés

«Événements d'instrumentation», à la page 10

Un événement d'instrumentation est une combinaison logique de conditions qu'un gestionnaire de files d'attente ou une instance de canal détecte et insère un message spécial, appelé *message d'événement*, dans une file d'attente d'événements.

[Utilisation des panneaux d'opérations et de contrôle sous z/OS](#)

### Tâches associées

[Automatisation des tâches d'administration](#)

[Utilisation des formats de commande programmables](#)

### Référence associée

«Types d'événements», à la page 12

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

## Les commandes MQSC

### Contrôle des événements du gestionnaire de files d'attente

Vous pouvez contrôler les événements du gestionnaire de files d'attente à l'aide des attributs du gestionnaire de files d'attente. Pour activer les événements du gestionnaire de files d'attente, définissez l'attribut de gestionnaire de files d'attente approprié sur ENABLED. Pour désactiver les événements du gestionnaire de files d'attente, définissez l'attribut de gestionnaire de files d'attente approprié sur DISABLED.


Pour activer ou désactiver des événements de gestionnaire de files d'attente, utilisez la commande MQSC **ALTER QMGR** en spécifiant l'attribut de gestionnaire de files d'attente approprié. Le [Tableau 1](#), à la page 21 récapitule comment activer les événements du gestionnaire de files d'attente. Pour désactiver un événement de gestionnaire de files d'attente, définissez le paramètre approprié sur DISABLED.

Événement	ALTER QMGR, paramètre
Droits d'accès	AUTHOREV (ENABLED)
Bloquer	INHIBTEV (ACTIVÉ)
Locale	LOCALEV (ACTIVÉ)
Eloigné	REMOTEEV (ACTIVÉ)
Démarrage et arrêt	STRSTPEV (ENABLED)

### Contrôle des événements de canal et de pont

Vous contrôlez les événements de canal à l'aide des attributs de gestionnaire de files d'attente. Pour activer les événements de canal, définissez l'attribut de gestionnaire de files d'attente approprié sur ENABLED. Pour désactiver les événements de canal, définissez l'attribut de gestionnaire de files d'attente approprié sur DISABLED.

Pour activer ou désactiver des événements de canal, utilisez la commande MQSC **ALTER QMGR** en spécifiant l'attribut de gestionnaire de files d'attente approprié. Le [Tableau 2](#), à la page 21 récapitule la façon dont vous activez les événements de canal et de pont. Pour désactiver un événement de gestionnaire de files d'attente, définissez le paramètre approprié sur DISABLED.

**Restriction :**  Les événements de définition automatique de canal ne sont pas disponibles sur IBM MQ for z/OS.

Événement	ALTER QMGR, paramètre
Canal	CHLEV (ACTIVÉ)
Lié aux erreurs de canal uniquement	CHLEV (EXCEPTION)
Passerelle IMS	BRIDGEEV (ACTIVÉ)
SSL	SSLEV (ACTIVÉ)
Définition automatique de canal	CHADEV (ENABLED)

Avec CHLEV défini sur exception, les codes retour suivants et les qualificatifs de raison correspondants sont générés:

- MQRC\_CHANNEL\_ACTIVATED
- MQRC\_CHANNEL\_CONV\_ERROR
- MQRC\_CHANNEL\_NOT\_ACTIVATED
- MQRC\_CHANNEL\_STOPPED
  - avec les ReasonQualifiers suivants:
    - MQRQ\_CHANNEL\_STOPPED\_ERROR

- MQRQ\_CHANNEL\_STOPPED\_RETRY
- MQRQ\_CHANNEL\_STOPPED\_DISABLED
- MQRQ\_CHANNEL\_STOPPED\_BY\_USER
- MQRQ\_CHANNEL\_BLOQUE
  - avec les ReasonQualifiers suivants:
    - MQRQ\_CHANNEL\_BLOCKED\_NOACCESS
    - MQRQ\_CHANNEL\_BLOCKED\_USERID
    - ADRESSE\_BLOC\_CANAL\_MQRQ\_CANAL

#### *Contrôle des événements de performances*

Vous contrôlez les événements de performances à l'aide de l'attribut de gestionnaire de files d'attente PERFMEV. Pour activer les événements de performances, définissez PERFMEV sur ENABLED. Pour désactiver les événements de performances, définissez l'attribut de gestionnaire de files d'attente PERFMEV sur DISABLED.

Pour définir l'attribut de gestionnaire de files d'attente PERFMEV sur ENABLED, utilisez la commande MQSC suivante:

```
ALTER QMGR PERFMEV (ENABLED)
```

Pour activer des événements de performances spécifiques, définissez l'attribut de file d'attente approprié. Spécifiez également les conditions à l'origine de l'événement.

#### **Événements de longueur de file d'attente**

Par défaut, tous les événements de longueur de file d'attente sont désactivés. Pour configurer une file d'attente pour l'un des événements de longueur de file d'attente:

1. Activez les événements de performances sur le gestionnaire de files d'attente.
2. Activez l'événement dans la file d'attente requise.
3. Définissez les limites, si nécessaire, sur les niveaux appropriés, exprimés en pourcentage de la longueur maximale de la file d'attente.

#### **Événements Intervalle de maintenance de file d'attente**

Pour configurer une file d'attente pour les événements d'intervalle de service de file d'attente, vous devez:

1. Activez les événements de performances sur le gestionnaire de files d'attente.
2. Définissez l'attribut de contrôle pour un événement Intervalle de service de file d'attente élevé ou OK sur la file d'attente, selon les besoins.
3. Indiquez l'intervalle de service en définissant l'attribut QSVICINT pour la file d'attente sur la durée appropriée.

**Remarque :** Lorsque cette option est activée, un événement d'intervalle de service de file d'attente peut être généré à tout moment approprié, n'attendant pas nécessairement l'émission d'un appel MQI pour la file d'attente. Toutefois, si un appel MQI est utilisé dans une file d'attente pour insérer ou supprimer un message, tout événement de performance applicable est généré à ce moment. L'événement n'est pas généré lorsque le temps écoulé devient égal à l'intervalle de service.

#### *Contrôle des événements de configuration, de commande et de consignateur*

Vous contrôlez les événements de configuration, de commande et de consignateur à l'aide des attributs de gestionnaire de files d'attente CONFIGEV, CMDEV et LOGGEREV. Pour activer ces événements, définissez l'attribut de gestionnaire de files d'attente approprié sur ENABLED. Pour désactiver ces événements, définissez l'attribut de gestionnaire de files d'attente approprié sur DISABLED.

## Événements de configuration

Pour activer les événements de configuration, définissez CONFIGEV sur ENABLED. Pour désactiver les événements de configuration, définissez CONFIGEV sur DISABLED. Par exemple, vous pouvez activer des événements de configuration à l'aide de la commande MQSC suivante:

```
ALTER QMGR CONFIGEV (ENABLED)
```

## Événements Commande

Pour activer les événements de commande, définissez CMDEV sur ENABLED. Pour activer les événements de commande pour les commandes à l'exception des commandes DISPLAY MQSC et des commandes Inquire PCF, définissez le paramètre CMDEV sur NODISPLAY. Pour désactiver les événements de commande, définissez CMDEV sur DISABLED. Par exemple, vous pouvez activer des événements de commande à l'aide de la commande MQSC suivante:

```
ALTER QMGR CMDEV (ENABLED)
```

## Événements de journalisation

Pour activer les événements du consignateur, définissez LOGGEREV sur ENABLED. Pour désactiver les événements du consignateur, définissez LOGGEREV sur DISABLED. Par exemple, vous pouvez activer des événements de consignateur à l'aide de la commande MQSC suivante:

```
ALTER QMGR LOGGEREV(ENABLED)
```

## Files d'attente d'événements

Lorsqu'un événement se produit, le gestionnaire de files d'attente place un message d'événement dans la file d'attente d'événements définie. Le message d'événement contient des informations sur l'événement.

Vous pouvez définir des files d'attente d'événements, comme suit:

- Files d'attente locales
- Files d'attente alias
- Définitions locales des files d'attente distantes ou en tant que
- Files d'attente de cluster éloignées

Si vous définissez toutes vos files d'attente d'événements en tant que définitions locales de la même file d'attente éloignée sur un gestionnaire de files d'attente, vous pouvez centraliser vos activités de surveillance.

Vous ne devez pas définir les files d'attente d'événements en tant que files d'attente de transmission, car les messages d'événement ont des formats incompatibles avec le format de message requis pour les files d'attente de transmission.

Les files d'attente d'événements partagées sont des files d'attente locales définies avec la valeur QSGDISP (SHARED).

Pour plus d'informations sur la définition de files d'attente partagées sur z/OS, voir [Programmation d'applications avec des files d'attente partagées](#).

## Lorsqu'une file d'attente d'événements n'est pas disponible

Si un événement se produit lorsque la file d'attente d'événements n'est pas disponible, le message d'événement est perdu. Par exemple, si vous ne définissez pas de file d'attente d'événements pour une catégorie d'événement, tous les messages d'événement de cette catégorie sont perdus. Les messages d'événement ne sont pas, par exemple, sauvegardés dans la file d'attente des messages non livrés.

Toutefois, vous pouvez définir la file d'attente d'événements en tant que file d'attente éloignée. Ensuite, en cas de problème sur le système distant lors de l'insertion de messages dans la file d'attente résolue, le message d'événement arrive dans la file d'attente des messages non livrés du système distant.

Une file d'attente d'événements peut être indisponible pour de nombreuses raisons, notamment:

- La file d'attente n'a pas été définie.
- La file d'attente a été supprimée.
- La file d'attente est saturée.
- L'insertion de la file d'attente a été interdite.

L'absence d'une file d'attente d'événements n'empêche pas l'événement de se produire. Par exemple, après un événement de performances, le gestionnaire de files d'attente modifie les attributs de file d'attente et réinitialise les statistiques de file d'attente. Cette modification se produit que le message d'événement soit inséré ou non dans la file d'attente des événements de performances. Il en va de même pour les événements de configuration et de commande.

## Utilisation de files d'attente d'événements déclenchés

Vous pouvez configurer les files d'attente d'événements avec des déclencheurs de sorte que lorsqu'un événement est généré, le message d'événement inséré dans la file d'attente d'événements démarre une application de surveillance écrite par l'utilisateur. Cette application peut traiter les messages d'événement et prendre les mesures appropriées. Par exemple, certains événements peuvent nécessiter l'information d'un opérateur, d'autres événements peuvent démarrer une application qui exécute automatiquement certaines tâches d'administration.

Les files d'attente d'événements peuvent être associées à des actions de déclenchement et peuvent créer des messages de déclenchement. Toutefois, si ces messages déclenchent à leur tour des conditions qui génèrent normalement un événement, aucun événement n'est généré. Le fait de ne pas générer d'événement dans cette instance garantit que la boucle ne se produit pas.

### Concepts associés

«Contrôle des événements», à la page 20

Vous activez et désactivez des événements en spécifiant les valeurs appropriées pour le gestionnaire de files d'attente, les attributs de file d'attente ou les deux, en fonction du type d'événement.

«Format des messages d'événement», à la page 24

Les messages d'événement contiennent des informations sur un événement et sa cause. Comme les autres messages IBM MQ, un message d'événement comporte deux parties: un descripteur de message et les données de message.

[Programmation d'application avec des files d'attente partagées](#)

[Conditions d'un événement déclencheur](#)

### Référence associée

[QSGDisp \(MQLONG\)](#)

### Format des messages d'événement

Les messages d'événement contiennent des informations sur un événement et sa cause. Comme les autres messages IBM MQ, un message d'événement comporte deux parties: un descripteur de message et les données de message.

- Le descripteur de message est basé sur la structure MQMD.
- Les données de message se composent d'un *en-tête d'événement* et des *données d'événement*. L'en-tête d'événement contient le code anomalie qui identifie le type d'événement. L'insertion du message d'événement, ainsi que toute action ultérieure, n'affecte pas le code anomalie renvoyé par l'appel MQI à l'origine de l'événement. Les données d'événement fournissent des informations supplémentaires sur l'événement.

En règle générale, vous traitez les messages d'événement à l'aide d'une application de gestion de système personnalisée pour répondre aux exigences de l'entreprise dans laquelle elle s'exécute.

Lorsque les gestionnaires de files d'attente d'un groupe de partage de files d'attente détectent les conditions de génération d'un message d'événement, plusieurs gestionnaires de files d'attente peuvent générer un message d'événement pour la file d'attente partagée, ce qui génère plusieurs messages



d'événement. Pour vous assurer qu'un système peut corrélérer plusieurs messages d'événement provenant de différents gestionnaires de files d'attente, ces messages d'événement ont un identificateur de corrélation unique (*CorrelId*) défini dans le descripteur de message (MQMD).

### **Référence associée**

[«Rapport d'activité MQMD \(descripteur de message\)», à la page 106](#)

Cette page permet d'afficher les valeurs contenues dans la structure MQMD d'un rapport d'activité

[«Rapport d'activité MQEPH \(en-tête PCF imbriqué\)», à la page 110](#)

Cette page permet d'afficher les valeurs contenues dans la structure MQEPH pour un rapport d'activité.

[«Rapport d'activité MQCFH \(en-tête PCF\)», à la page 111](#)

Cette page permet d'afficher les valeurs PCF contenues dans la structure MQCFH d'un rapport d'activité.

[Référence de message d'événement](#)

[Format de message d'événement](#)

[MQMD de message d'événement \(descripteur de message\)](#)

[Message d'événement MQCFH \(en-tête PCF\)](#)

[Descriptions des messages d'événement](#)

## **Événements Performances**

Les événements de performances sont liés à des conditions qui peuvent affecter les performances des applications qui utilisent une file d'attente spécifiée. La portée des événements de performances est la file d'attente. Les appels **MQPUT** et **MQGET** sur une file d'attente n'affectent pas la génération d'événements de performances sur une autre file d'attente.

Les messages d'événement de performances peuvent être générés à n'importe quel moment, mais ils n'attendent pas nécessairement qu'un appel MQI soit émis pour la file d'attente. Toutefois, si vous utilisez un appel MQI sur une file d'attente pour insérer ou supprimer un message, tous les événements de performances appropriés sont générés à ce moment-là.

Chaque message d'événement de performance généré est placé dans la file d'attente, SYSTEM.ADMIN.PERFM.EVENT.

Les données d'événement contiennent un code anomalie qui identifie la cause de l'événement, un ensemble de statistiques d'événement de performance et d'autres données. Les types de données d'événement pouvant être renvoyés dans les messages d'événement de performances sont décrits dans la liste suivante:

- [Longueur élevée de file d'attente](#)
- [Longueur faible de file d'attente](#)
- [File d'attente saturée](#)
- [Intervalle de service de file d'attente important](#)
- [Intervalle de service de file d'attente OK](#)

Les exemples illustrant l'utilisation des événements de performances supposent que vous définissez des attributs de file d'attente à l'aide des commandes IBM MQ appropriées (MQSC). Sous z/OS, vous pouvez également définir des attributs de file d'attente à l'aide des panneaux d'opérations et de contrôle des gestionnaires de files d'attente.

### **Référence associée**

[«Types d'événements», à la page 12](#)

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.


## **Statistiques d'événement de performance**

Les données d'événement de performances du message d'événement contiennent des statistiques sur l'événement. Utilisez les statistiques pour analyser le comportement d'une file d'attente spécifiée.

Les données d'événement du message d'événement contiennent des informations sur l'événement pour les programmes de gestion du système. Pour tous les événements de performances, les données d'événement contiennent les noms du gestionnaire de files d'attente et la file d'attente associée à l'événement. Les données d'événement contiennent également des statistiques relatives à l'événement. Le Tableau 3, à la page 26 récapitule les statistiques d'événement que vous pouvez utiliser pour analyser le comportement d'une file d'attente. Toutes les statistiques font référence à ce qui s'est passé depuis la dernière réinitialisation des statistiques.

<i>Tableau 3. Statistiques d'événement de performance</i>	
<b>Paramètre</b>	<b>Description</b>
TimeSinceReset	Temps écoulé depuis la dernière réinitialisation des statistiques.
HighQDepth	Nombre maximal de messages dans la file d'attente depuis la dernière réinitialisation des statistiques.
MsgEnqCount	Nombre de messages mis en file d'attente (nombre d'appels MQPUT à la file d'attente) depuis la dernière réinitialisation des statistiques.
MsgDeqCount	Nombre de messages retirés de la file d'attente (nombre d'appels MQGET à la file d'attente) depuis la dernière réinitialisation des statistiques.

Les statistiques d'événement de performance sont réinitialisées lorsque l'une des modifications suivantes se produit:

- Un événement de performance se produit (les statistiques sont réinitialisées sur tous les gestionnaires de files d'attente actifs).
- Un gestionnaire de files d'attente s'arrête et redémarre.
- La commande PCF, Réinitialiser les statistiques de file d'attente, est émise à partir d'un programme d'application.
-  Sous z/OS uniquement, la commande RESET QSTATS est émise sur la console.

### **Concepts associés**

«Événements Performances», à la page 25

Les événements de performances sont liés à des conditions qui peuvent affecter les performances des applications qui utilisent une file d'attente spécifiée. La portée des événements de performances est la file d'attente. Les appels **MQPUT** et **MQGET** sur une file d'attente n'affectent pas la génération d'événements de performances sur une autre file d'attente.

«Le temporisateur de service», à la page 28

Les événements d'intervalle de service de file d'attente utilisent un temporisateur interne, appelé *temporisateur de service*, qui est contrôlé par le gestionnaire de files d'attente. Le temporisateur de service est utilisé uniquement si un événement d'intervalle de service de file d'attente est activé.

«Règles pour les événements d'intervalle de service de file d'attente», à la page 29

Les règles formelles contrôlent le moment où le temporisateur de service est défini et où les événements d'intervalle de service de file d'attente sont générés.

### **Tâches associées**

«Activation des événements d'intervalle de service de file d'attente», à la page 29

Pour configurer une file d'attente pour les événements d'intervalle de service de file d'attente, vous devez définir le gestionnaire de files d'attente et les attributs de file d'attente appropriés.

### **Référence associée**

[Longueur élevée de file d'attente](#)

[Réinitialiser les statistiques de file d'attente](#)

[RESET QSTATS](#)

## **Événements Intervalle de maintenance de file d'attente**

Les événements d'intervalle de service de file d'attente indiquent si une opération a été effectuée sur une file d'attente dans un intervalle de temps défini par l'utilisateur appelé *intervalle de service*. En fonction de votre installation, vous pouvez utiliser des événements d'intervalle de service de file d'attente pour surveiller si les messages sont supprimés des files d'attente assez rapidement.

Les événements d'intervalle de service de file d'attente ne sont pas pris en charge sur les files d'attente partagées.

Les types d'événement d'intervalle de service de file d'attente suivants peuvent se produire, où le terme *opération d'obtention* fait référence à un appel **MQGET** ou à une activité qui supprime un message d'une file d'attente, par exemple à l'aide de la commande **CLEAR QLOCAL** :

### **Intervalle de service de file d'attente OK**

Indique qu'après l'une des opérations suivantes:

- Un appel MQPUT
- Une opération d'extraction qui laisse une file d'attente non vide

une opération d'extraction a été effectuée au cours d'une période définie par l'utilisateur, appelée *intervalle de service*.

Seule une opération d'extraction peut provoquer le message d'événement Intervalle de service de file d'attente OK. Les événements d'intervalle de service de file d'attente OK sont parfois décrits comme des événements OK.

### **Intervalle de service de file d'attente important**

Indique qu'après l'une des opérations suivantes:

- Un appel MQPUT
- Une opération d'extraction qui laisse une file d'attente non vide

une opération d'extraction n'a **pas** été effectuée dans un intervalle de service défini par l'utilisateur.

Une opération d'extraction ou un appel MQPUT peut provoquer le message d'événement Intervalle élevé du service de file d'attente. Les événements d'intervalle élevé du service de file d'attente sont parfois décrits comme des événements élevés.

Pour activer à la fois les événements Intervalle de service de file d'attente OK et Intervalle de service de file d'attente élevé, définissez l'attribut de contrôle QServiceIntervalEvent sur Elevé. Les événements d'intervalle de service de file d'attente OK sont automatiquement activés lorsqu'un événement d'intervalle de service de file d'attente élevé est généré. Il n'est pas nécessaire d'activer les événements d'intervalle de service de file d'attente OK indépendamment.

Les événements OK et High étant mutuellement exclusifs, si l'un est activé, l'autre est désactivé. Toutefois, les deux événements peuvent être désactivés simultanément.

La [Figure 3](#), à la page 28 présente un graphique de la longueur de la file d'attente par rapport au temps. A l'heure P1, une application émet une instruction MQPUT pour placer un message dans la file d'attente. A l'heure G1, une autre application émet une commande MQGET pour supprimer le message de la file d'attente.

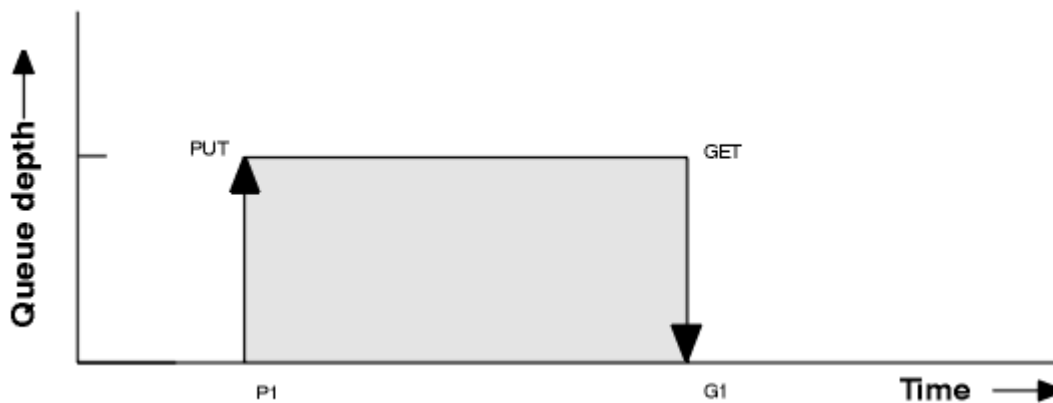


Figure 3. Description des événements d'intervalle de service de file d'attente

Les résultats possibles des événements d'intervalle de service de file d'attente sont les suivants:

- Si le temps écoulé entre l'insertion et l'extraction est inférieur ou égal à l'intervalle de service:
  - Un événement *Intervalle de service de file d'attente OK* est généré à l'heure G1, si les événements d'intervalle de service de file d'attente sont activés
- Si le temps écoulé entre l'insertion et l'extraction est supérieur à l'intervalle de service:
  - Un événement *Intervalle de service de file d'attente élevé* est généré à l'heure G1, si les événements d'intervalle de service de file d'attente sont activés.

L'algorithme de démarrage du temporisateur de service et de génération d'événements est décrit dans «[Règles pour les événements d'intervalle de service de file d'attente](#)», à la page 29.

#### Référence associée

[Intervalle de service de file d'attente OK](#)

[Intervalle de service de file d'attente important](#)

[Événement QServiceInterval\(MQLONG\)](#)

[Événement QServiceInterval\(entier signé à 10 chiffres\)](#)

#### Le temporisateur de service

Les événements d'intervalle de service de file d'attente utilisent un temporisateur interne, appelé *temporisateur de service*, qui est contrôlé par le gestionnaire de files d'attente. Le temporisateur de service est utilisé uniquement si un événement d'intervalle de service de file d'attente est activé.

#### Que mesure précisément le minuteur de service?

Le temporisateur de service mesure le temps écoulé entre un appel MQPUT vers une file d'attente vide ou une opération d'extraction et l'insertion ou l'extraction suivante, à condition que la longueur de la file d'attente soit différente de zéro entre ces deux opérations.

#### Quand le temporisateur de service est-il actif?

Le temporisateur de service est toujours actif (en cours d'exécution), si la file d'attente contient des messages (la longueur est différente de zéro) et qu'un événement d'intervalle de service de file d'attente est activé. Si la file d'attente devient vide (longueur de file d'attente zéro), le temporisateur passe à l'état OFF, pour être redémarré lors de l'insertion suivante.

#### Quand le temporisateur de service est-il réinitialisé?

Le temporisateur de service est toujours réinitialisé après une opération d'extraction. Il est également réinitialisé par un appel MQPUT à une file d'attente vide. Cependant, il n'est pas nécessairement réinitialisé sur un événement d'intervalle de service de file d'attente.

#### Comment le temporisateur de service est-il utilisé?

À la suite d'une opération d'extraction ou d'un appel MQPUT, le gestionnaire de files d'attente compare le temps écoulé mesuré par le temporisateur de service avec l'intervalle de service défini par l'utilisateur. Le résultat de cette comparaison est que:

- Un événement OK est généré s'il existe une opération d'extraction et que le temps écoulé est inférieur ou égal à l'intervalle de service, ET que cet événement est activé.
- Un événement élevé est généré si le temps écoulé est supérieur à l'intervalle de service, ET que cet événement est activé.

### **Les applications peuvent-elles lire le temporisateur de service?**

Non, le temporisateur de service est un temporisateur interne qui n'est pas disponible pour les applications.

### **Qu'en est-il du paramètre *TimeSinceReset* ?**

Le paramètre *TimeSinceReset* est renvoyé dans le cadre des statistiques d'événement dans les données d'événement. Il indique le délai entre les événements d'intervalle de service de file d'attente successifs, sauf si les statistiques d'événement sont réinitialisées.

### *Règles pour les événements d'intervalle de service de file d'attente*

Les règles formelles contrôlent le moment où le temporisateur de service est défini et où les événements d'intervalle de service de file d'attente sont générés.

## **Règles pour le temporisateur de service**

Le temporisateur de service est remis à zéro et redémarré comme suit:

- Après un appel MQPUT à une file d'attente vide.
- Après un appel MQGET, si la file d'attente n'est pas vide après l'appel MQGET.

La réinitialisation du temporisateur ne dépend pas de la génération ou non d'un événement.

Au démarrage du gestionnaire de files d'attente, le temporisateur de service est défini sur l'heure de démarrage si la longueur de la file d'attente est supérieure à zéro.

Si la file d'attente est vide suite à une opération d'extraction, le temporisateur passe à l'état OFF.

## **Événements d'intervalle élevé du service de file d'attente**

L'événement Intervalle de service de file d'attente doit être activé (défini sur HIGH).

Les événements d'intervalle de service de file d'attente élevé sont automatiquement activés lorsqu'un événement d'intervalle de service de file d'attente OK est généré.

Si le temps de service est supérieur à l'intervalle de service, un événement est généré le ou avant la prochaine opération MQPUT ou d'extraction.

## **Événements d'intervalle de service de file d'attente OK**

Les événements d'intervalle de service de file d'attente OK sont automatiquement activés lorsqu'un événement d'intervalle de service de file d'attente élevé est généré.

Si le temps de service (temps écoulé) est inférieur ou égal à l'intervalle de service, un événement est généré le ou avant l'opération d'extraction suivante.

### **Tâches associées**

«Activation des événements d'intervalle de service de file d'attente», à la page 29

Pour configurer une file d'attente pour les événements d'intervalle de service de file d'attente, vous devez définir le gestionnaire de files d'attente et les attributs de file d'attente appropriés.

### *Activation des événements d'intervalle de service de file d'attente*

Pour configurer une file d'attente pour les événements d'intervalle de service de file d'attente, vous devez définir le gestionnaire de files d'attente et les attributs de file d'attente appropriés.

## Pourquoi et quand exécuter cette tâche

Les événements élevés et OK s'excluent mutuellement, c'est-à-dire que lorsque l'un est activé, l'autre est automatiquement désactivé:

- Lorsqu'un événement élevé est généré dans une file d'attente, le gestionnaire de files d'attente désactive automatiquement les événements élevés et active les événements OK pour cette file d'attente.
- Lorsqu'un événement OK est généré dans une file d'attente, le gestionnaire de files d'attente désactive automatiquement les événements OK et active les événements élevés pour cette file d'attente.

Événement Intervalle de service de file d'attente	Attributs File d'attente
Intervalle de service de file d'attente important Intervalle de service de file d'attente OK Aucun événement d'intervalle de service de file d'attente	QSVCI EV (ELEV E) QSVCI EV (OK) QSVCI EV (NONE)
Intervalle de maintenance	QSVCI NT ( <i>tt</i> ) où <i>tt</i> est le service temps d'intervalle en millisecondes.

Pour activer les événements d'intervalle de service de file d'attente, procédez comme suit:

### Procédure

1. Définissez l'attribut de gestionnaire de files d'attente **PERFMEV** sur ENABLED.  
Les événements de performances sont activés sur le gestionnaire de files d'attente.
2. Définissez l'attribut de contrôle, **QSVCI EV**, pour un événement Intervalle de service de file d'attente élevé ou OK dans la file d'attente, selon les besoins.
3. Définissez l'attribut **QSVCI NT** pour la file d'attente afin de spécifier l'intervalle de service approprié.

### Exemple

Pour activer les événements d'intervalle de service de file d'attente élevé avec un intervalle de service de 10 secondes (10 000 millisecondes), utilisez les commandes MQSC suivantes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QSVCI NT(10000) QSVCI EV(HIGH)
```

### Exemples d'événements d'intervalle de service de file d'attente

Utilisez les exemples de cette section pour comprendre les informations que vous pouvez obtenir à partir des événements d'intervalle de service de file d'attente.

Les trois exemples de sous-rubriques fournissent des illustrations de plus en plus complexes de l'utilisation des événements d'intervalle de service de file d'attente.

Les figures accompagnant les exemples de chaque sous-rubrique ont la même structure:

- La figure 1 est un graphique de la longueur de la file d'attente en fonction du temps, montrant les appels MQGET et MQPUT individuels.
- La section Commentaire présente une comparaison des contraintes de temps. Vous devez tenir compte de trois périodes:
  - Intervalle de service défini par l'utilisateur.

- Temps mesuré par le temporisateur de service.
- Temps écoulé depuis la dernière réinitialisation des statistiques d'événement (TimeSinceReset dans les données d'événement).
- La section Récapitulatif des statistiques d'événement indique quels événements sont activés à tout moment et quels événements sont générés.

Les exemples illustrent les aspects suivants des événements d'intervalle de service de file d'attente:

- Variation de la longueur de la file d'attente dans le temps.
- Comparaison du temps écoulé mesuré par le temporisateur de service avec l'intervalle de service.
- Quel événement est activé.
- Les événements qui sont générés.

**A faire :** L'exemple 1 illustre un cas simple où les messages sont intermittents et où chaque message est supprimé de la file d'attente avant l'arrivée du message suivant. D'après les données d'événement, vous savez que le nombre maximal de messages dans la file d'attente était de un. Vous pouvez donc déterminer la durée pendant laquelle chaque message se trouvait dans la file d'attente.

Toutefois, dans le cas général, lorsqu'il existe plusieurs messages dans la file d'attente et que la séquence des appels MQGET et MQPUT n'est pas prévisible, vous ne pouvez pas utiliser les événements d'intervalle de service de file d'attente pour calculer la durée de conservation d'un message individuel dans une file d'attente. Le paramètre de réinitialisation TimeSince, qui est renvoyé dans les données d'événement, peut inclure une proportion de temps lorsqu'il n'y a pas de messages dans la file d'attente. Par conséquent, les résultats que vous dérivez à partir de ces statistiques sont implicitement moyennés pour inclure ces temps.

### Concepts associés

«Événements Intervalle de maintenance de file d'attente», à la page 27

Les événements d'intervalle de service de file d'attente indiquent si une opération a été effectuée sur une file d'attente dans un intervalle de temps défini par l'utilisateur appelé *intervalle de service*. En fonction de votre installation, vous pouvez utiliser des événements d'intervalle de service de file d'attente pour surveiller si les messages sont supprimés des files d'attente assez rapidement.

«Le temporisateur de service», à la page 28

Les événements d'intervalle de service de file d'attente utilisent un temporisateur interne, appelé *temporisateur de service*, qui est contrôlé par le gestionnaire de files d'attente. Le temporisateur de service est utilisé uniquement si un événement d'intervalle de service de file d'attente est activé.

*Événements d'intervalle de service de file d'attente: exemple 1*

Séquence de base des appels MQGET et MQPUT, où la longueur de la file d'attente est toujours égale à un ou zéro.

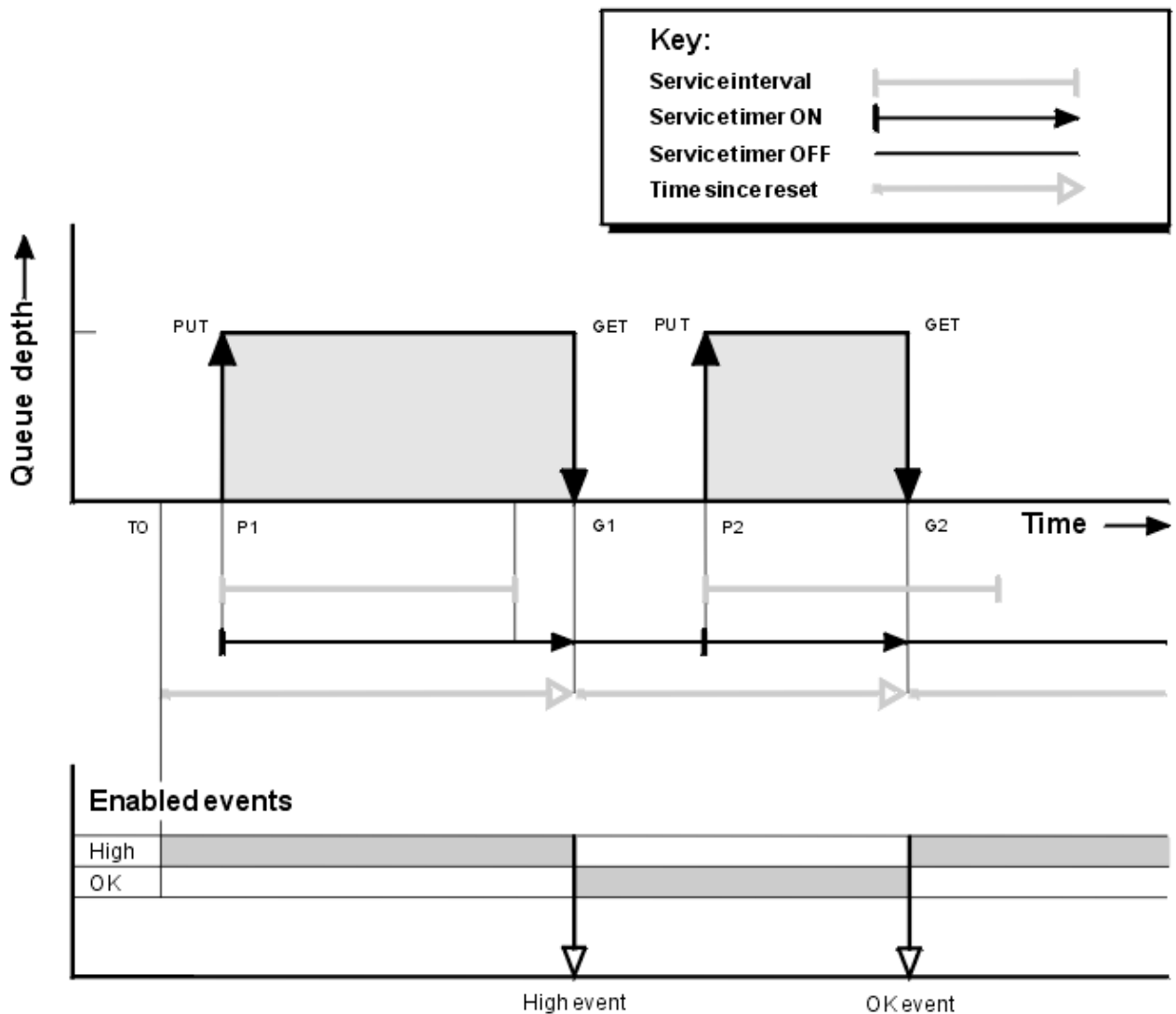


Figure 4. Événements d'intervalle de service de file d'attente-Exemple 1

## Commentaire

1. Dans P1, une application place un message dans une file d'attente vide. Cette opération démarre le temporisateur de service.

Notez que T0 peut être l'heure de démarrage du gestionnaire de files d'attente.

2. Au niveau G1, une autre application extrait le message de la file d'attente. Etant donné que le temps écoulé entre P1 et G1 est supérieur à l'intervalle de service, un événement Intervalle de service de file d'attente élevé est généré dans l'appel MQGET à la valeur G1. Lorsque l'événement élevé est généré, le gestionnaire de files d'attente réinitialise l'attribut de contrôle d'événement de sorte que:
  - a. L'événement OK est automatiquement activé.
  - b. L'événement élevé est désactivé.

Etant donné que la file d'attente est vide, le temporisateur de service passe à l'état OFF.

3. A l'étape P2, un deuxième message est inséré dans la file d'attente. Cette opération redémarre le temporisateur de service.
4. Dans G2, le message est supprimé de la file d'attente. Toutefois, comme le temps écoulé entre P2 et G2 est inférieur à l'intervalle de service, un événement d'intervalle de service de file d'attente OK est généré sur l'appel MQGET à l'adresse G2. Lorsque l'événement OK est généré, le gestionnaire de files d'attente réinitialise l'attribut de contrôle de sorte que:



- a. L'événement élevé est automatiquement activé.
- b. L'événement OK est désactivé.

La file d'attente étant vide, le temporisateur de service passe à nouveau à l'état OFF.

## Récapitulatif des statistiques d'événement

Le [Tableau 5](#), à la [page 33](#) récapitule les statistiques d'événement de cet exemple.

<i>Tableau 5. Récapitulatif des statistiques d'événement pour l'exemple 1</i>		
<b>Attribut</b>	<b>Événement 1</b>	<b>Événement 2</b>
Heure de l'événement	T (G1)	T (G2)
Type d'événement	Elevé	OK
TimeSinceReset	T (G1)-T (0)	T (G2)-T (G1)
HighQDepth	1	1
MsgEnqCount	1	1
MsgDeqCount	1	1

La partie centrale de [Figure 4](#), à la [page 32](#) indique le temps écoulé, mesuré par le temporisateur de service, par rapport à l'intervalle de service de cette file d'attente. Pour voir si un événement d'intervalle de service de file d'attente peut se produire, comparez la longueur de la ligne horizontale représentant le temporisateur de service (avec une flèche) à celle de la ligne représentant l'intervalle de service. Si la ligne du temporisateur de service est plus longue et que l'événement Intervalle de service de file d'attente élevé est activé, un événement Intervalle de service de file d'attente élevé se produit lors de la prochaine extraction. Si la ligne de temporisation est plus courte et que l'événement Intervalle de service de file d'attente OK est activé, un événement Intervalle de service de file d'attente OK se produit lors de la prochaine extraction.

### *Événements d'intervalle de service de file d'attente: exemple 2*

Séquence d'appels MQPUT et d'appels MQGET, dans laquelle la longueur de la file d'attente n'est pas toujours égale à un ou à zéro.

Cet exemple montre également des instances du temporisateur réinitialisé sans que des événements soient générés, par exemple, à l'heure P2.

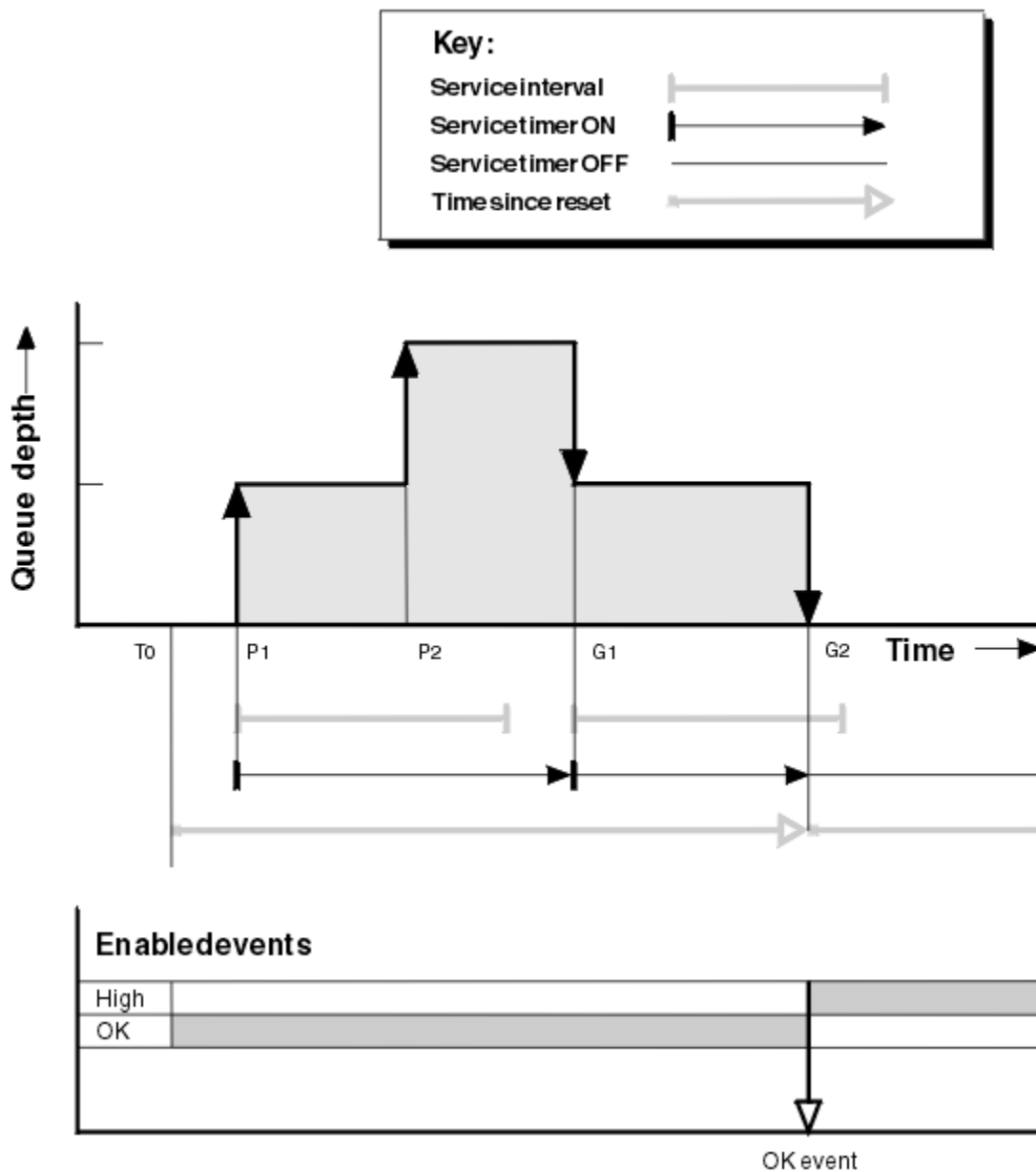


Figure 5. Événements d'intervalle de service de file d'attente-Exemple 2

### Commentaire

Dans cet exemple, les événements OK sont activés initialement et les statistiques de file d'attente ont été réinitialisées à l'heure T0.

1. Sur P1, la première insertion démarre le temporisateur de service.
2. En P2, la seconde insertion ne génère pas d'événement car une insertion ne peut pas provoquer d'événement OK.
3. A la G1, l'intervalle de maintenance a été dépassé et par conséquent, un événement OK n'est pas généré. Toutefois, l'appel MQGET entraîne la réinitialisation du temporisateur de service.
4. Lors de la G2, la seconde extraction se produit dans l'intervalle de service et, cette fois, un événement OK est généré. Le gestionnaire de files d'attente réinitialise l'attribut de contrôle d'événement de sorte que:
  - a. L'événement élevé est automatiquement activé.
  - b. L'événement OK est désactivé.

Etant donné que la file d'attente est vide, le temporisateur de service passe à l'état OFF.

## Récapitulatif des statistiques d'événement

Le Tableau 6, à la page 35 récapitule les statistiques d'événement de cet exemple.

Attribut	Événement 2
Heure de l'événement	T (G2)
Type d'événement	OK
TimeSinceReset	T (G2)-T (0)
HighQDepth	2
MsgEnqCount	2
MsgDeqCount	2

### Événements d'intervalle de service de file d'attente: exemple 3

Séquence d'appels MQGET et d'appels MQPUT plus sporadique que les exemples précédents.

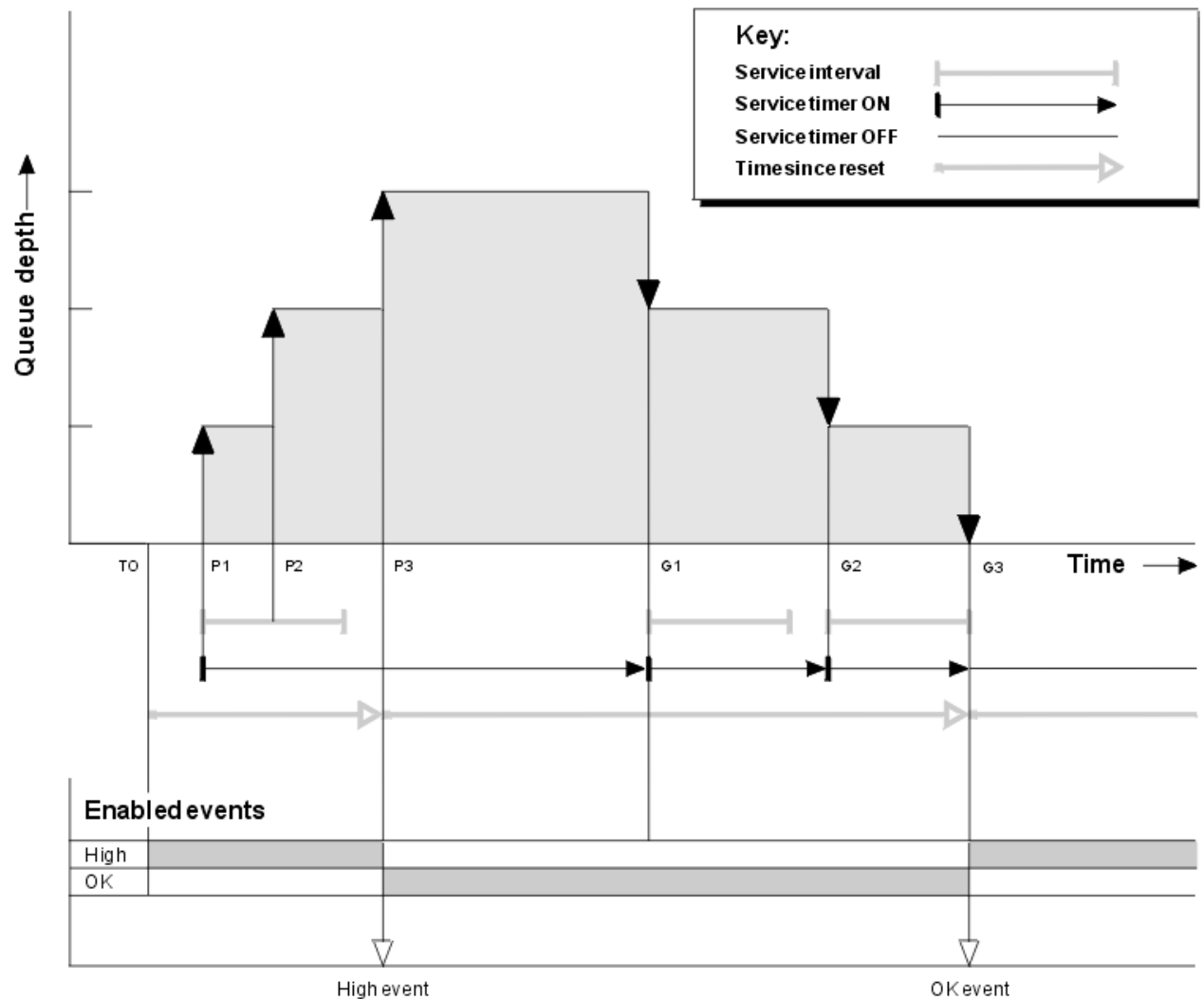


Figure 6. Événements d'intervalle de service de file d'attente-Exemple 3

## Commentaire

1. A l'heure T (0), les statistiques de file d'attente sont réinitialisées et les événements d'intervalle élevé du service de file d'attente sont activés.
2. Sur P1, la première insertion démarre le temporisateur de service.
3. En P2, la seconde insertion augmente la longueur de la file d'attente à deux. Un événement élevé n'est pas généré ici car l'intervalle de service n'a pas été dépassé.
4. A P3, la troisième insertion entraîne la génération d'un événement élevé. (Le temporisateur a dépassé l'intervalle de service.) Le temporisateur n'est pas réinitialisé car la longueur de la file d'attente n'était pas égale à zéro avant l'insertion. Toutefois, les événements OK sont activés.
5. Au niveau G1, l'appel MQGET ne génère pas d'événement car l'intervalle de service a été dépassé et les événements OK sont activés. L'appel MQGET réinitialise toutefois le temporisateur de service.
6. Au niveau G2, l'appel MQGET ne génère pas d'événement car l'intervalle de service a été dépassé et les événements OK sont activés. A nouveau, l'appel MQGET réinitialise le temporisateur de service.
7. A G3, la troisième extraction vide la file d'attente et le temporisateur de service est égal à l'intervalle de service. Par conséquent, un événement OK est généré. Le temporisateur de service est réinitialisé et les événements élevés sont activés. L'appel MQGET vide la file d'attente, ce qui place le temporisateur à l'état OFF.

## Récapitulatif des statistiques d'événement

Le Tableau 7, à la page 36 récapitule les statistiques d'événement de cet exemple.

Attribut	Événement 1	Événement 2
Heure de l'événement	T (P3)	T (G3)
Type d'événement	Elevé	OK
TimeSinceReset	T (P3)-T (0)	T (G3)-T (P3)
HighQDepth	3	3
MsgEnqCount	3	0
MsgDeqCount	0	3

### ***Événements de longueur de file d'attente***

Les événements de longueur de file d'attente sont liés à la longueur de la file d'attente, c'est-à-dire au nombre de messages dans la file d'attente.

Dans les applications IBM MQ, les files d'attente ne doivent pas être saturées. Dans ce cas, les applications ne peuvent plus placer de messages dans la file d'attente qu'elles spécifient. Bien que le message ne soit pas perdu si cela se produit, une file d'attente saturée peut entraîner des inconvénients considérables. Le nombre de messages peut s'accumuler dans une file d'attente si les messages sont placés dans la file d'attente plus rapidement que les applications qui les traitent peuvent les retirer.

La solution à ce problème dépend des circonstances particulières, mais peut impliquer:

- Détournement de certains messages vers une autre file d'attente.
- Démarrage de nouvelles applications pour retirer plus de messages de la file d'attente.
- Arrêt du trafic des messages non essentiels.
- Augmentation de la longueur de la file d'attente pour dépasser un maximum transitoire.

L'avertissement à l'avance que les problèmes peuvent être en cours de route facilite la prise de mesures préventives. A cet effet, IBM MQ fournit les événements de longueur de file d'attente suivants:

### Événements Longueur élevée

Indique que la longueur de la file d'attente a augmenté jusqu'à un seuil prédéfini appelé Limite supérieure de la longueur de la file d'attente.

### Événements Longueur faible

Indique que la longueur de la file d'attente a été réduite à un seuil prédéfini appelé limite inférieure de la longueur de la file d'attente.

### Événements de file d'attente saturée

Indique que la file d'attente a atteint sa longueur maximale, c'est-à-dire que la file d'attente est saturée.

Un événement de file d'attente saturée est généré lorsqu'une application tente d'insérer un message dans une file d'attente qui a atteint sa longueur maximale. Les événements Longueur élevée de la file d'attente avertissent à l'avance qu'une file d'attente est en passe de se remplir. Cela signifie qu'après avoir reçu cet événement, l'administrateur système doit prendre des mesures préventives. Vous pouvez configurer le gestionnaire de files d'attente de sorte que, si l'action préventive aboutit et que le nombre de lignes de la file d'attente passe à un niveau plus sûr, le gestionnaire de files d'attente génère un événement Longueur faible de la file d'attente.

Le premier exemple d'événement de longueur de file d'attente illustre l'effet d'une action présumée empêchant la saturation de la file d'attente.

### Concepts associés

[«Exemples d'événements de longueur de file d'attente»](#), à la page 40

Utilisez ces exemples pour comprendre les informations que vous pouvez obtenir à partir des événements de longueur de file d'attente

### Référence associée

[File d'attente saturée](#)

[Longueur élevée de file d'attente](#)



[Longueur faible de file d'attente](#)

### Activation des événements de longueur de file d'attente

Pour configurer une file d'attente pour l'un des événements de longueur de file d'attente, vous devez définir le gestionnaire de files d'attente et les attributs de file d'attente appropriés.

### Pourquoi et quand exécuter cette tâche

Par défaut, tous les événements de longueur de file d'attente sont désactivés. Lorsque cette option est activée, les événements de longueur de file d'attente sont générés comme suit:

- Un événement Longueur élevée de file d'attente est généré lorsqu'un message est inséré dans la file d'attente, entraînant une longueur de file d'attente supérieure ou égale à la valeur définie pour **QDepthHighLimit**.
  - Un événement Longueur élevée de file d'attente est automatiquement activé par un événement Longueur faible de file d'attente dans la même file d'attente.
  - Un événement Longueur élevée de file d'attente active automatiquement un événement Longueur faible de file d'attente et un événement File d'attente saturée sur la même file d'attente.
-  **Multi** Un événement Longueur faible de file d'attente est généré lorsqu'un message est supprimé d'une file d'attente par une opération GET, ce qui entraîne une longueur de file d'attente inférieure ou égale à la valeur définie pour **QDepthLowLimit**.
-  **z/OS** Un événement Longueur minimale de la file d'attente est généré lorsqu'un message est supprimé d'une file d'attente par une opération GET ou qu'il aurait été supprimé mais qu'il est arrivé à expiration, ce qui a pour conséquence que la longueur de la file d'attente est inférieure ou égale à la valeur définie pour **QDepthLowLimit**.

- Un événement Longueur faible de file d'attente est automatiquement activé par un événement Longueur élevée de file d'attente ou par un événement File d'attente saturée dans la même file d'attente.
- Un événement Longueur faible de la file d'attente active automatiquement un événement Longueur élevée de la file d'attente et un événement File d'attente saturée dans la même file d'attente.
- Un événement File d'attente saturée est généré lorsqu'une application ne parvient pas à placer un message dans une file d'attente car la file d'attente est saturée.
  - Un événement File d'attente saturée est automatiquement activé par un événement Longueur élevée de file d'attente ou Longueur faible de file d'attente dans la même file d'attente.
  - Un événement File d'attente saturée active automatiquement un événement Longueur faible de la file d'attente sur la même file d'attente.

Pour configurer une file d'attente pour l'un des événements de longueur de file d'attente, procédez comme suit:

## Procédure

1. Activez les événements de performances sur le gestionnaire de files d'attente à l'aide de l'attribut de gestionnaire de files d'attente **PERFMEV**.

Les événements sont placés dans la file d'attente `SYSTEM.ADMIN.PERFM.EVENT`.

2. Définissez l'un des attributs suivants pour activer l'événement dans la file d'attente requise:

- *ÉvénementQDepthHigh* (**QDPHIEV** dans MQSC)
- *ÉvénementQDepthLow* (**QDPLOEV** dans MQSC)
- *ÉvénementQDepthMax* (**QDPMAXEV** dans MQSC)

3. Facultatif : Pour définir les limites, affectez les attributs suivants, sous forme de pourcentage de la longueur maximale de la file d'attente:

- *QDepthHigh* (**QDEPTHHI** dans MQSC)
- *QDepthLowLimite* (**QDEPTHLO** dans MQSC)

**Restriction : QDEPTHHI** ne doit pas être inférieur à **QDEPTHLO**.

Si **QDEPTHHI** est égal à **QDEPTHLO**, un message d'événement est généré chaque fois que la longueur de la file d'attente dépasse la valeur dans les deux sens, car le seuil haut est activé lorsque la longueur de la file d'attente est inférieure à la valeur et le seuil bas est activé lorsque la longueur est supérieure à la valeur.

## Résultats

### Remarque :

**Multi** Un événement Longueur faible de la file d'attente n'est pas généré lorsque des messages arrivés à expiration sont supprimés d'une file d'attente par une opération GET entraînant une longueur de file d'attente inférieure ou égale à la valeur définie pour **QDepthLowLimit**. IBM MQ génère le message d'événement de longueur faible de la file d'attente uniquement lors d'une opération GET réussie. Par conséquent, lorsque les messages arrivés à expiration sont supprimés de la file d'attente, aucun message d'événement de longueur faible de la file d'attente n'est généré. En outre, après la suppression de ces messages arrivés à expiration de la file d'attente, *QDepthHighEvent* et *QDepthLowEvent* ne sont pas réinitialisés.

**z/OS** IBM MQ génère le message d'événement de longueur faible de la file d'attente au cours d'une opération GET destructive ou d'une opération GET destructive qui aurait abouti si un message correspondant n'avait pas expiré. Sinon, lorsque des messages arrivés à expiration sont supprimés d'une file d'attente lors du traitement d'arrière-plan général, aucun message d'événement bas de longueur de file d'attente n'est généré. En outre, après la suppression des messages arrivés à expiration d'une file d'attente lors du traitement d'arrière-plan général, *QDepthHighEvent* et *QDepthLowEvent* ne sont

pas réinitialisés. Pour plus d'informations sur le traitement des messages arrivés à expiration, voir [Optimisation de votre gestionnaire de files d'attente sur IBM MQ for z/OS](#).

## Exemple

Pour activer les événements Longueur élevée de file d'attente dans la file d'attente MYQUEUE avec une limite définie à 80%, utilisez les commandes MQSC suivantes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHHI(80) QDPHIEV(ENABLED)
```

Pour activer les événements Longueur faible de la file d'attente dans la file d'attente MYQUEUE avec une limite définie à 20%, utilisez les commandes MQSC suivantes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDEPTHLO(20) QDPLOEV(ENABLED)
```

Pour activer les événements File d'attente saturée dans la file d'attente MYQUEUE, utilisez les commandes MQSC suivantes:

```
ALTER QMGR PERFMEV(ENABLED)
ALTER QLOCAL('MYQUEUE') QDPMAEV(ENABLED)
```

## Shared queues and queue depth events on z/OS

On IBM MQ for z/OS, event monitoring is more straightforward for an application that uses shared queues if all the queue managers in the queue sharing group have the same setting for the **PERFMEV** attribute.

When a queue depth event occurs on a shared queue, and the queue manager attribute **PERFMEV** is set to **ENABLED**, the queue managers in the queue sharing group produce an event message. If **PERFMEV** is set to **DISABLED** on some of the queue managers, event messages are not produced by those queue managers, making event monitoring from an application more difficult. For more straightforward monitoring, give each queue manager the same setting for the **PERFMEV** attribute.

This event message that each queue manager generates represents its individual usage of the shared queue. If a queue manager performs no activity on the shared queue, various values in the event message are null or zero. You can use null event messages as follows:

- Ensure that each active queue manager in a queue sharing group generates one event message
- Highlight cases of no activity on a shared queue for the queue manager that produced the event message

## Coordinating queue manager

When a queue manager issues a queue depth event, it updates the shared queue object definition to toggle the active performance event attributes. For example, depending on the definition of the queue attributes, a Queue Depth High event enables a Queue Depth Low and a Queue Full event. After updating the shared queue object successfully, the queue manager that detected the performance event initially becomes the *coordinating queue manager*.

If enabled for performance events, the coordinating queue manager performs the following actions:

1. Issues an event message that captures all shared queue performance data it has gathered since the last time an event message was created, or since the queue statistics were last reset. The message descriptor (MQMD) of this message contains a unique correlation identifier (*CorrelId*) created by the coordinating queue manager.

2. Broadcasts to all other *active* queue managers in the same queue sharing group to request the production of an event message for the shared queue. The broadcast contains the correlation identifier created by the coordinating queue manager for the set of event messages.

Having received a request from the coordinating queue manager, if there is an active queue manager in the queue sharing group that is enabled for performance events, that active queue manager issues an event message for the shared queue. The event message that is issued contains information about all the operations performed by the receiving (active) queue manager since the last time an event message was created, or since the statistics were last reset. The message descriptor (MQMD) of this event message contains the unique correlation identifier (*CorrelId*) specified by the coordinating queue manager.

When performance events occur on a shared queue,  $n$  event messages are produced, where  $n$  is a number from 1 to the number of active queue managers in the queue sharing group. Each event message contains data that relates to the shared queue activity for the queue manager that generated the event message.

## Differences between shared and nonshared queues

Enabling queue depth events on shared queues differs from enabling them on nonshared queues. A key difference is that events are enabled for shared queues even if **PERFMEV** is DISABLED on the queue manager. This is not the case for nonshared queues.

Consider the following example, which illustrates this difference:

- QM1 is a queue manager with *PerformanceEvent* (**PERFMEV** in MQSC) set to DISABLED.
- SQ1 is a shared queue with **QSGDISP** set to (SHARED) QLOCAL in MQSC.
- LQ1 is a nonshared queue with **QSGDISP** set to (QMGR) QLOCAL in MQSC.

Both queues have the following attributes set on their definitions:

- **QDPHIEV (ENABLED)**
- **QDPLOEV (DISABLED)**
- **QDPMAXEV (DISABLED)**

If messages are placed on both queues so that the depth meets or exceeds the **QDEPTHHI** threshold, the **QDPHIEV** value on SQ1 switches to DISABLED. Also, **QDPLOEV** and **QDPMAXEV** are switched to ENABLED. SQ1's attributes are automatically switched for each performance event at the time the event criteria are met.

In contrast the attributes for LQ1 remain unchanged until **PERFMEV** on the queue manager is ENABLED. This means that if, for example, the queue manager's **PERFMEV** attribute is ENABLED, DISABLED and then set to ENABLED again, the performance event settings on shared queues might not be consistent with those of nonshared queues, even though they might have initially been the same.

## Exemples d'événements de longueur de file d'attente

Utilisez ces exemples pour comprendre les informations que vous pouvez obtenir à partir des événements de longueur de file d'attente

Le premier exemple fournit une illustration de base des événements de longueur de file d'attente. Le deuxième exemple est plus étendu, mais les principes sont les mêmes que pour le premier exemple. Les deux exemples utilisent la même définition de file d'attente, comme suit:

La longueur maximale de la file d'attente, MYQUEUE1, est de 1000 messages. La limite de longueur maximale de la file d'attente est de 80% et la limite de longueur minimale de la file d'attente est de 20%. Initialement, les événements Longueur élevée de la file d'attente sont activés, tandis que les autres événements Longueur élevée de la file d'attente sont désactivés.

Les commandes IBM MQ (MQSC) permettant de configurer cette file d'attente sont les suivantes:

```
ALTER QMGR PERFMEV(ENABLED)

DEFINE QLOCAL('MYQUEUE1') MAXDEPTH(1000) QDPMAXEV(DISABLED) QDEPTHHI(80)
QDPHIEV(ENABLED) QDEPTHLO(20) QDPLOEV(DISABLED)
```



## Concepts associés

«Événements de longueur de file d'attente», à la page 36

Les événements de longueur de file d'attente sont liés à la longueur de la file d'attente, c'est-à-dire au nombre de messages dans la file d'attente.

## Tâches associées

«Activation des événements de longueur de file d'attente», à la page 37

Pour configurer une file d'attente pour l'un des événements de longueur de file d'attente, vous devez définir le gestionnaire de files d'attente et les attributs de file d'attente appropriés.

## Référence associée

Les commandes MQSC

*Événements de longueur de file d'attente: exemple 1*

Séquence de base des événements de longueur de file d'attente.

La Figure 7, à la page 41 présente la variation de la longueur de la file d'attente dans le temps.

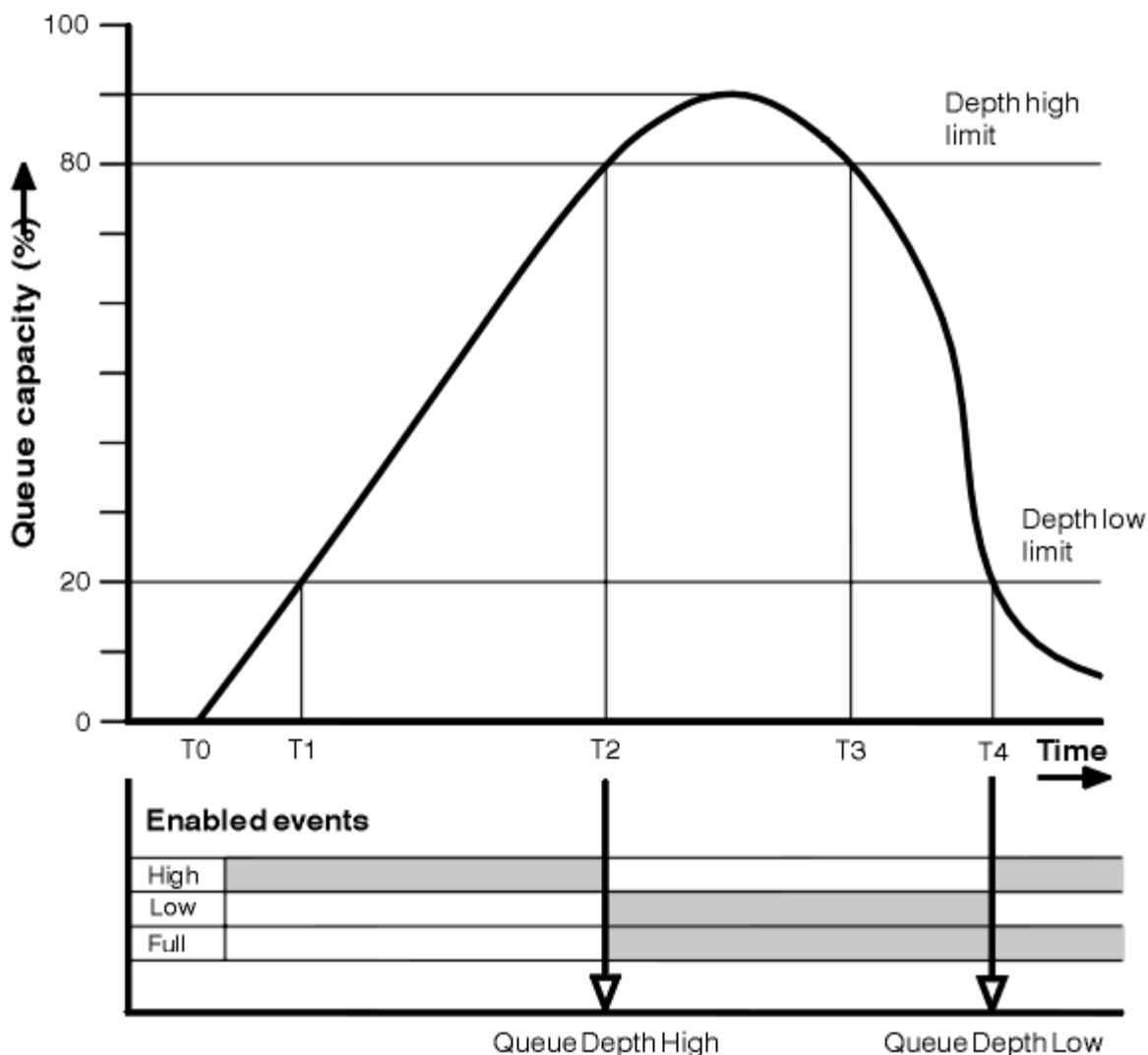


Figure 7. Événements de longueur de file d'attente (1)

## Commentaire

1. A T (1), la longueur de la file d'attente augmente (plus d'appels MQPUT que d'appels MQGET) et dépasse la limite inférieure de la longueur de la file d'attente. Aucun événement n'est généré pour le moment.

2. La longueur de la file d'attente continue d'augmenter jusqu'à T (2), lorsque la limite supérieure de longueur (80%) est atteinte et qu'un événement Longueur élevée de file d'attente est généré.  
Cela active les événements File d'attente saturée et Longueur faible de la file d'attente.
3. Les actions préventives (présumées) lancées par l'événement empêchent la saturation de la file d'attente. Par le temps T (3), la limite supérieure de la longueur de la file d'attente a été atteinte à nouveau, cette fois à partir du haut. Aucun événement n'est généré pour le moment.
4. La longueur de la file d'attente continue de tomber jusqu'à T (4), lorsqu'elle atteint la limite inférieure de longueur (20%) et qu'un événement Longueur faible de la file d'attente est généré.  
Cela active les événements File d'attente saturée et Longueur élevée de la file d'attente.

## Récapitulatif des statistiques d'événement

Tableau 8, à la page 42 récapitule les statistiques d'événements de file d'attente et [Tableau 9, à la page 42](#) récapitule les événements qui sont activés.

*Tableau 8. Récapitulatif des statistiques d'événement pour les événements de longueur de file d'attente (exemple 1)*

Statistiques d'événement	Événement 2	Événement 4
Heure de l'événement	T (2)	T (4)
Type d'événement	Longueur élevée de file d'attente	Longueur faible de file d'attente
TimeSinceReset	T (2)-T (0)	T (4)-T (2)
HighQDepth (Longueur maximale de la file d'attente depuis la réinitialisation)	800	900
MsgEnqCount	1157	1220
MsgDeqCount	357	1820

*Tableau 9. Récapitulatif indiquant les événements activés*

Plage de temps	Événement Longueur élevée de la file d'attente	Événement Longueur faible de la file d'attente	Événement File d'attente saturée
Avant T (1)	Activée	-	-
T (1) à T (2)	Activée	-	-
T (2) à T (3)	-	Activée	Activée
T (3) à T (4)	-	Activée	Activée
Après T (4)	Activée	-	Activée

*Événements de longueur de file d'attente: exemple 2*

Une séquence plus étendue d'événements de longueur de file d'attente.

La [Figure 8, à la page 43](#) présente la variation de la longueur de la file d'attente dans le temps.

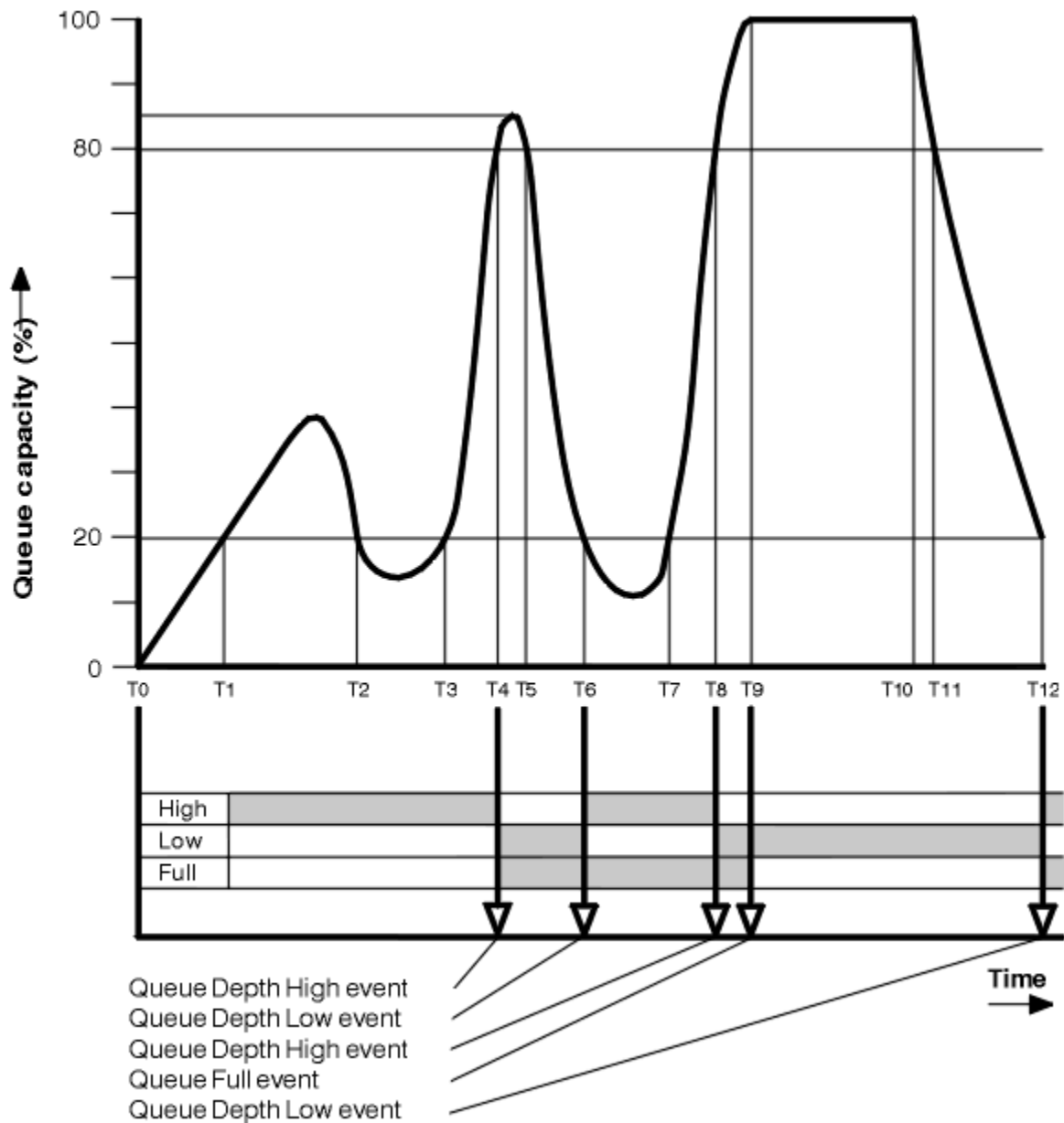


Figure 8. Evénements de longueur de file d'attente (2)

### Commentaire

1. Aucun événement Longueur faible de la file d'attente n'est généré aux moments suivants:
  - T (1) (La profondeur de la file d'attente augmente et n'est pas activée)
  - T (2) (Non activé)
  - T (3) (augmentation de la longueur de la file d'attente et non activé)
2. A T (4), un événement Longueur élevée de file d'attente se produit. Cela active les événements File d'attente saturée et Longueur faible de la file d'attente.
3. A T (9), un événement File d'attente saturée se produit **après** le premier message qui ne peut pas être inséré dans la file d'attente car la file d'attente est saturée.
4. A T (12), un événement Longueur faible de la file d'attente se produit.

## Récapitulatif des statistiques d'événement

Le [Tableau 10](#), à la page 44 récapitule les statistiques d'événements de file d'attente et le [Tableau 11](#), à la page 44 récapitule les événements qui sont activés à des moments différents pour cet exemple.

*Tableau 10. Récapitulatif des statistiques d'événement pour les événements de longueur de file d'attente (exemple 2)*

Statistiques d'événement	Événement 4	Événement 6	Événement 8	Événement 9	Événement 12
Heure de l'événement	T (4)	T (6)	T (8)	T (9)	T (12)
Type d'événement	Longueur élevée de file d'attente	Longueur faible de file d'attente	Longueur élevée de file d'attente	File d'attente saturée	Longueur faible de file d'attente
TimeSinceReset	T (4)-T (0)	T (6)-T (4)	T (8)-T (6)	T (9)-T (8)	T (12)-T (9)
HighQDepth	800	855	800	1000	1000
MsgEnqCount	1645	311	1377	324	221
MsgDeqCount	845	911	777	124	1021

*Tableau 11. Récapitulatif indiquant les événements activés*

Plage de temps	Événement Longueur élevée de la file d'attente	Événement Longueur faible de la file d'attente	Événement File d'attente saturée
T (0) à T (4)	Activée	-	-
T (4) à T (6)	-	Activée	Activée
T (6) à T (8)	Activée	-	Activée
T (8) à T (9)	-	Activée	Activée
T (9) à T (12)	-	Activée	-
Après T (12)	Activée	-	Activée

**Remarque :** Les événements ne sont pas synchronisés. Par conséquent, vous pouvez avoir une file d'attente vide, puis la remplir à l'origine d'un événement, puis annuler tous les messages sous le contrôle d'un gestionnaire de points de synchronisation. Toutefois, l'activation des événements a été définie automatiquement, de sorte que lors de la prochaine saturation de la file d'attente, aucun événement n'est généré.

## Événements de configuration

Les événements de configuration sont des notifications qui sont générées lorsqu'un objet est créé, modifié ou supprimé et qui peuvent également être générées par des demandes explicites.

Pour plus d'informations sur la façon dont vous activez les événements, voir [«Contrôle des événements de configuration, de commande et de consignateur»](#), à la page 22 .

Les événements de configuration vous informent des modifications apportées aux attributs d'un objet. Il existe quatre types d'événements de configuration:

- Créer des événements d'objet
- Événements de modification d'objet
- Événements de suppression d'objet

- Actualiser les événements d'objet

Les données d'événement contiennent les informations suivantes:

#### **Informations d'origine**

comprend le gestionnaire de files d'attente à partir duquel la modification a été effectuée, l'ID de l'utilisateur qui a effectué la modification et la manière dont la modification a été apportée, par exemple par une commande de console.

#### **Informations contextuelles**

une réplique des informations de contexte dans les données de message du message de commande.

Les informations contextuelles sont incluses dans les données d'événement uniquement lorsque la commande a été entrée en tant que message sur SYSTEM.COMMAND.INPUT d'entrée.

#### **Identité de l'objet**

comprend le nom, le type et la disposition de l'objet.

#### **Attributs d'objet**

comprend les valeurs de tous les attributs de l'objet.

L'événement est un message au format PCF. Pour plus d'informations, voir :

- [Modifier un objet](#)
- [Créer un objet](#)
- [Supprimer l'objet](#)
- [Régénérer l'objet](#)

Dans le cas d'événements d'objet de modification, un message avant et un message après sont générés. Ils ont en commun les zones suivantes, afin que vous puissiez identifier les messages correspondants:

- Le même ID de corrélation
- Dans l'en-tête PCF:
  - L'en-tête de message antérieur comporte `MsgSeqNumber' : 1, 'Control' : 'NOT_LAST'`
  - L'en-tête de message après comporte `MsgSeqNumber' : 2, 'Control' : 'LAST'`

Chaque message d'événement de configuration généré est placé dans la file d'attente SYSTEM.ADMIN.CONFIG.EVENT.

#### **Concepts associés**

«Événements de configuration», à la page 17

Les événements de configuration sont générés lorsqu'un événement de configuration est demandé explicitement ou automatiquement lorsqu'un objet est créé, modifié ou supprimé.

#### **Référence associée**

[Créer un objet](#)

[Modifier un objet](#)

[Supprimer l'objet](#)

[Actualiser l'objet](#)

«Types d'événements», à la page 12

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

#### **Génération d'événements de configuration**

Cette page permet d'afficher les commandes qui entraînent la génération d'événements de configuration et de comprendre les circonstances dans lesquelles les événements de configuration ne sont pas générés.

Un message d'événement de configuration est inséré dans la file d'attente d'événements de configuration lorsque l'attribut de gestionnaire de files d'attente CONFIGEV est défini sur ENABLED et

- L'une des commandes suivantes, ou son équivalent PCF, est émise:

- DELETE AUTHINFO
- DELETE CFSTRUCT
- Supprimer le canal
- Supprimer une liste de noms
- Supprimer un processus
- SUPPRIMER QMODEL/QALIAS/QREMOTE
- DELETE STGCLASS
- Supprimer la rubrique
- ACTUALISEZ LE GESTIONNAIRE DE FILES D'ATTENTE
- Les commandes suivantes, ou leur équivalent PCF, sont émises même si aucune modification n'est apportée à l'objet:
  - DEFINE/ALTER AUTHINFO
  - DEFINE/ALTER CFSTRUCT
  - DEFINIR/MODIFIER LE CANAL
  - DEFINE/ALTER NOMELIST
  - DEFINE/ALTER PROCESSUS
  - DEFINIR/MODIFIER QMODEL/QALIAS/QREMOTE
  - DEFINE/ALTER STGCLASS
  - DEFINE/ALTER TOPIC
  - DEFINE MAXSMSGS
  - SET CHLAUTH
  - ALTER QMGR, sauf si l'attribut CONFIGEV est DISABLED et n'est pas remplacé par ENABLED
- Les commandes suivantes, ou leur équivalent PCF, sont émises pour une file d'attente locale qui n'est pas une file d'attente dynamique temporaire, même si aucune modification n'est apportée à la file d'attente.
  - SUPPRIMER QLOCAL
  - DEFINE/ALTER QLOCAL
- un appel MQSET est émis, autre que pour une file d'attente dynamique temporaire, même si l'objet n'est pas modifié.

### **Lorsque des événements de configuration ne sont pas générés**

Les messages d'événements de configuration ne sont pas générés dans les cas suivants:

- Lorsqu'une commande ou un appel MQSET échoue
- Lorsqu'un gestionnaire de files d'attente rencontre une erreur lors de la tentative d'insertion d'un événement de configuration dans la file d'attente d'événements, auquel cas la commande ou l'appel MQSET se termine, mais aucun message d'événement n'est généré
- Pour une file d'attente dynamique temporaire
- Lorsque des modifications internes sont apportées à l'attribut de file d'attente TRIGGER
- Pour la file d'attente d'événements de configuration SYSTEM.ADMIN.CONFIG.EVENT, sauf par la commande REFRESH QMGR
- Pour les commandes REFRESH/RESET CLUSTER et RESUME/SUSPEND QMGR qui entraînent des modifications de la mise en cluster
- Lors de la création ou de la suppression d'un gestionnaire de files d'attente

### **Concepts associés**

Présentation des commandes PCF (Programmable Command Formats)

«Événements de configuration», à la page 44

Les événements de configuration sont des notifications qui sont générées lorsqu'un objet est créé, modifié ou supprimé et qui peuvent également être générées par des demandes explicites.

### **Référence associée**

[Les commandes MQSC](#)

[MQSET-Définition des attributs d'objet](#)

[MQSET-Définition des attributs d'objet](#)

### **Utilisation de l'événement de configuration**

Utilisez cette page pour voir comment vous pouvez utiliser les événements de configuration pour obtenir des informations sur votre système et pour comprendre les facteurs, tels que CMDSCOPE, qui peuvent affecter votre utilisation des événements de configuration.

Vous pouvez utiliser des événements de configuration dans les buts suivants:

1. Pour produire et gérer un référentiel de configuration central, à partir duquel des rapports peuvent être produits et des informations sur la structure du système peuvent être générées.
2. Pour générer une trace d'audit. Par exemple, si un objet est modifié de manière inattendue, les informations relatives à l'auteur de la modification et au moment où elle a été effectuée peuvent être stockées.

Cela peut être particulièrement utile lorsque des événements de commande sont également activés. Si une commande MQSC ou PCF entraîne la génération d'un événement de configuration et d'un événement de commande, les deux messages d'événement partagent le même identificateur de corrélation dans leur descripteur de message.

Pour un appel MQSET ou l'une des commandes suivantes:

- DEFINE, objet
- Objet ALTER
- Supprimer l'objet

si l'attribut de gestionnaire de files d'attente CONFIGEV est activé, mais que le message d'événement de configuration ne peut pas être inséré dans la file d'attente d'événements de configuration, par exemple si la file d'attente d'événements n'a pas été définie, la commande ou l'appel MQSET est exécuté indépendamment.

### **Effets du CMDSCOPE**

Pour les commandes dans lesquelles CMDSCOPE est utilisé, le ou les messages d'événement de configuration sont générés sur le ou les gestionnaires de files d'attente dans lesquels la commande est exécutée, et non dans lesquels la commande est entrée. Toutefois, toutes les informations d'origine et de contexte dans les données d'événement seront liées à la commande d'origine telle qu'elle a été entrée, même si la commande utilisant CMDSCOPE est une commande qui a été générée par le gestionnaire de files d'attente source.

Lorsqu'un groupe de partage de files d'attente inclut des gestionnaires de files d'attente qui ne sont pas à la version en cours, des événements sont générés pour toute commande exécutée à l'aide de CMDSCOPE sur un gestionnaire de files d'attente qui est à la version en cours, mais pas sur ceux qui sont à la version précédente. Cela se produit même si le gestionnaire de files d'attente dans lequel la commande est entrée est à la version précédente, bien que dans ce cas, aucune information de contexte ne soit incluse dans les données d'événement.

### **Concepts associés**

[Présentation des commandes PCF \(Programmable Command Formats\)](#)

«Événements de configuration», à la page 44

Les événements de configuration sont des notifications qui sont générées lorsqu'un objet est créé, modifié ou supprimé et qui peuvent également être générées par des demandes explicites.

### **Référence associée**

[MQSET-Définition des attributs d'objet](#)

[MQSET-Définition des attributs d'objet](#)

### ***Événement d'actualisation de la configuration d'objet***

L'événement de configuration d'objet d'actualisation est différent des autres événements de configuration, car il se produit uniquement lorsqu'il est demandé explicitement.

Les événements de création, de modification et de suppression sont générés par un appel MQSET ou par une commande permettant de modifier un objet, mais l'événement d'actualisation d'objet se produit uniquement lorsqu'il est explicitement demandé par la commande MQSC, REFRESH QMGR ou son équivalent PCF.

La commande REFRESH QMGR est différente de toutes les autres commandes qui génèrent des événements de configuration. Toutes les autres commandes s'appliquent à un objet particulier et génèrent un événement de configuration unique pour cet objet. La commande REFRESH QMGR peut générer de nombreux messages d'événement de configuration représentant potentiellement chaque définition d'objet stockée par un gestionnaire de files d'attente. Un message d'événement est généré pour chaque objet sélectionné.

La commande REFRESH QMGR utilise une combinaison de trois critères de sélection pour filtrer le nombre d'objets impliqués:

- Nom de l'objet
- Type d'objet
- Fréquence de régénération

Si vous n'indiquez aucun critère de sélection dans la commande REFRESH QMGR, les valeurs par défaut sont utilisées pour chaque critère de sélection et un message d'événement de configuration d'actualisation est généré pour chaque définition d'objet stockée par le gestionnaire de files d'attente. Cela peut entraîner des temps de traitement inacceptables et la génération de messages d'événement. Envisagez de spécifier des critères de sélection.

La commande REFRESH QMGR qui génère les événements d'actualisation peut être utilisée dans les situations suivantes:

- Lorsque des données de configuration sont requises sur tout ou partie des objets d'un système, que les objets aient ou non été récemment manipulés, par exemple, lorsque les événements de configuration sont activés pour la première fois.

Envisagez d'utiliser plusieurs commandes, chacune avec une sélection d'objets différente, mais de sorte que toutes soient incluses.

- Si une erreur s'est produite dans SYSTEM.ADMIN.CONFIG.EVENT EVENT. Dans ce cas, aucun message d'événement de configuration n'est généré pour les événements Créer, Modifier ou Supprimer. Lorsque l'erreur dans la file d'attente a été corrigée, la commande Refresh Queue Manager peut être utilisée pour demander la génération de messages d'événement qui ont été perdus alors qu'une erreur s'est produite dans la file d'attente. Dans cette situation, envisagez de définir l'intervalle d'actualisation sur l'heure à laquelle la file d'attente n'est pas disponible.

### **Concepts associés**

«Événements de configuration», à la page 44

Les événements de configuration sont des notifications qui sont générées lorsqu'un objet est créé, modifié ou supprimé et qui peuvent également être générées par des demandes explicites.

### **Référence associée**

[ACTUALISEZ LE GESTIONNAIRE DE FILES D'ATTENTE](#)

[Régénérer un gestionnaire de files d'attente](#)



## Événements Commande

Les événements de commande sont des notifications indiquant qu'une commande MQSC ou PCF a été exécutée avec succès.

Les données d'événement contiennent les informations suivantes:

### Informations d'origine

comprend le gestionnaire de files d'attente à partir duquel la commande a été émise, l'ID de l'utilisateur qui a émis la commande et la manière dont la commande a été émise, par exemple par une commande de console.

### Informations contextuelles

une réplique des informations de contexte dans les données de message du message de commande. Si aucune commande n'est entrée à l'aide d'un message, les informations de contexte sont omises.

Les informations contextuelles sont incluses dans les données d'événement uniquement lorsque la commande a été entrée en tant que message sur SYSTEM.COMMAND.INPUT d'entrée.

### Informations sur la commande

le type de commande qui a été émise.

### Données de commande

- pour les commandes PCF, une réplique des données de commande
- pour les commandes MQSC, le texte de la commande

Le format des données de la commande ne correspond pas nécessairement à celui de la commande d'origine. Par exemple, sous Multiplateformes, le format des données de commande est toujours au format PCF, même si la demande d'origine était une commande MQSC.

Chaque message d'événement de commande généré est placé dans la file d'attente d'événements de commande, SYSTEM.ADMIN.COMMAND.EVENT.

### Référence associée

#### Commande

«Types d'événements», à la page 12

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

### ***Génération d'événement de commande***

Cette page permet d'afficher les situations qui provoquent la génération d'événements de commande et de comprendre les circonstances dans lesquelles les événements de commande ne sont pas générés.

Un message d'événement de commande est généré dans les situations suivantes:

- Lorsque l'attribut de gestionnaire de files d'attente CMDEV est défini sur ENABLED et qu'une commande MQSC ou PCF s'exécute correctement.
- Lorsque l'attribut de gestionnaire de files d'attente CMDEV est défini sur NODISPLAY et qu'une commande s'exécute correctement, à l'exception des commandes DISPLAY (MQSC) et des commandes Inquire (PCF).
- Lorsque vous exécutez la commande MQSC, ALTER QMGR ou la commande PCF, le gestionnaire de files d'attente de modification et l'attribut de gestionnaire de files d'attente CMDEV remplissent l'une des conditions suivantes:
  - CMDEV n'est pas défini comme DISABLED après la modification
  - CMDEV n'a pas été indiqué comme DISABLED avant la modification

Si une commande est exécutée sur la file d'attente d'événements de commande, SYSTEM.ADMIN.COMMAND.EVENT, un événement de commande est généré si la file d'attente existe toujours et qu'elle n'est pas verrouillée.

## Lorsque des événements de commande ne sont pas générés

Un message d'événement de commande n'est pas généré dans les cas suivants:

- En cas d'échec d'une commande
- Lorsqu'un gestionnaire de files d'attente rencontre une erreur lors de la tentative d'insertion d'un événement de commande dans la file d'attente d'événements, auquel cas la commande s'exécute indépendamment, mais aucun message d'événement n'est généré
- Pour la commande MQSC REFRESH QMGR TYPE (EARLY)
- Pour la commande MQSC START QMGR MQSC
- Pour la commande MQSC SUSPEND QMGR, si le paramètre LOG est spécifié
- Pour la commande MQSC RESUME QMGR, si le paramètre LOG est spécifié

### Concepts associés

«Événements Commande», à la page 49

Les événements de commande sont des notifications indiquant qu'une commande MQSC ou PCF a été exécutée avec succès.

### Référence associée

[ACTUALISEZ LE GESTIONNAIRE DE FILES D'ATTENTE](#)

[START QMGR](#)

[SUSPEND QMGR](#)

[RESUME QMGR](#)

[SUSPEND QMGR, RESUME QMGR et clusters](#)

### Utilisation de l'événement de commande

Utilisez cette page pour voir comment vous pouvez utiliser les événements de commande pour générer une trace d'audit des commandes qui ont été exécutées

Par exemple, si un objet est modifié de manière inattendue, les informations relatives à l'auteur de la modification et au moment où elle a été effectuée peuvent être stockées. Cela peut s'avérer particulièrement utile lorsque des événements de configuration sont également activés. Si une commande MQSC ou PCF entraîne la génération d'un événement de commande et d'un événement de configuration, les deux messages d'événement partagent le même identificateur de corrélation dans leur descripteur de message.

Si un message d'événement de commande est généré, mais ne peut pas être placé dans la file d'attente d'événements de commande, par exemple si la file d'attente d'événements de commande n'a pas été définie, la commande pour laquelle l'événement de commande a été généré est toujours exécutée.

### Effets du CMDSCOPE

Pour les commandes dans lesquelles CMDSCOPE est utilisé, le ou les messages d'événement de commande sont générés sur le ou les gestionnaires de files d'attente dans lesquels la commande est exécutée, et non sur le ou les gestionnaires de files d'attente dans lesquels la commande est entrée. Toutefois, toutes les informations d'origine et de contexte dans les données d'événement seront liées à la commande d'origine telle qu'elle a été entrée, même si la commande utilisant CMDSCOPE est une commande qui a été générée par le gestionnaire de files d'attente source.

### Concepts associés

«Événements Commande», à la page 49

Les événements de commande sont des notifications indiquant qu'une commande MQSC ou PCF a été exécutée avec succès.

«Génération d'événement de commande», à la page 49

Cette page permet d'afficher les situations qui provoquent la génération d'événements de commande et de comprendre les circonstances dans lesquelles les événements de commande ne sont pas générés.

## Référence associée

[Les commandes MQSC](#)

[Commandes et réponses PCF dans les groupes](#)

Multi

## Événements de journalisation

Les événements du consignateur sont des notifications indiquant qu'un gestionnaire de files d'attente a commencé à écrire dans un nouveau domaine réservé aux journaux **IBM i** ou, sous IBM i, dans un récepteur de journal. **z/OS** Les messages d'événement du consignateur ne sont pas disponibles avec IBM MQ for z/OS.

Les données d'événement contiennent les informations suivantes:

- Nom du domaine réservé aux journaux en cours.
- Nom de l'extension de journal la plus ancienne requise pour la reprise par redémarrage.
- Nom de l'extension de journal la plus ancienne requise pour la reprise sur incident lié au support.
- Répertoire dans lequel se trouvent les extensions de journal.
- Nom de l'extension de journal la plus ancienne pour laquelle une notification d'archivage est requise.

Un événement de consignateur est généré pour la gestion des journaux d'archivage, c'est-à-dire lorsque **ARCHLOG** est modifié, et la valeur **ARCHLOG** est incluse dans le message d'événement de consignateur.

Pour plus d'informations sur tous ces paramètres, voir [DISPLAY QMSTATUS](#).

Vous pouvez utiliser les valeurs **CURRLOG** et **ARCHLOG** pour déterminer ce qui doit être archivé. Chaque fois que **CURRLOG** est modifié, les fichiers dont le nombre est inférieur à **CURRLOG** peuvent être envoyés pour archivage et, une fois l'archivage terminé pour l'extension, vous devez appeler [SET LOG](#) pour avertir le gestionnaire de files d'attente.

**ARCHLOG** est le domaine le plus ancien qui doit être archivé. Une fois que vous avez archivé ce domaine et appelé [SET LOG](#) pour informer le gestionnaire de files d'attente qu'il a été archivé, le gestionnaire de files d'attente déplace **ARCHLOG** vers le domaine suivant dans l'ordre. Le gestionnaire de files d'attente planifie la suppression ou la réutilisation des extensions antérieures à **ARCHLOG**.

Une fois que **SET LOG** a été appelé pour ce domaine, un nouvel événement est émis, car **ARCHLOG** a changé et vous devez savoir ce qui doit être archivé après le nouvel événement.

Si, pour une raison quelconque, votre processus d'archivage échoue et qu'une génération importante de notifications se produit, votre administrateur peut émettre manuellement la commande [RESET QMGR TYPE \(ARCHLOG\)](#). Le gestionnaire de files d'attente est alors informé qu'il peut réutiliser ou supprimer toutes les extensions plus anciennes que celles que vous avez spécifiées.

Chaque message d'événement du consignateur généré est placé dans la file d'attente d'événements du consignateur, SYSTEM.ADMIN.LOGGER.EVENT.

## Référence associée

[Consignateur](#)

«Types d'événements», à la page 12

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

Multi

## Génération d'événements de consignateur

Cette page permet d'afficher les situations qui entraînent la génération d'événements de consignateur et de comprendre les circonstances dans lesquelles les événements de consignateur ne sont pas générés.

Un message d'événement de consignateur est généré dans les situations suivantes:

- Lorsque l'attribut de gestionnaire de files d'attente LOGGEREV est défini sur ENABLED et que le gestionnaire de files d'attente commence à écrire dans un nouveau domaine réservé aux journaux ou, sous IBM i, dans un récepteur de journal.

- Lorsque l'attribut de gestionnaire de files d'attente LOGGERS est défini sur ENABLED et que le gestionnaire de files d'attente démarre.
- Lorsque l'attribut de gestionnaire de files d'attente LOGGERS passe de DISABLED à ENABLED.
- Lorsque l'attribut de gestionnaire de files d'attente LOGGERS est informé que le nom du domaine réservé au journal le plus ancien, pour lequel le gestionnaire de files d'attente attend une notification d'archivage (ARCHLOG), a été archivé.

**Conseil :** Vous pouvez utiliser la commande RESET QMGR MQSC pour demander à un gestionnaire de files d'attente de commencer à écrire dans un nouveau domaine réservé aux journaux.

## Lorsque les événements du consignateur ne sont pas générés

Un message d'événement de consignateur n'est pas généré dans les cas suivants:



- Lorsqu'un gestionnaire de files d'attente est configuré pour utiliser la journalisation avec réutilisation automatique des journaux.

Dans ce cas, l'attribut de gestionnaire de files d'attente LOGGERS est défini sur DISABLED et ne peut pas être modifié.

- Lorsqu'un gestionnaire de files d'attente rencontre une erreur lors de la tentative d'insertion d'un événement de consignateur dans la file d'attente d'événements, auquel cas l'action à l'origine de l'événement se termine, mais aucun message d'événement n'est généré.

### Concepts associés

«Événements de journalisation», à la page 51

Les événements du consignateur sont des notifications indiquant qu'un gestionnaire de files d'attente a commencé à écrire dans un nouveau domaine réservé aux journaux  ou, sous IBM i, dans un récepteur de journal.  Les messages d'événement du consignateur ne sont pas disponibles avec IBM MQ for z/OS.

### Référence associée

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(entier signé à 10 chiffres\)](#)

[RESET QMGR](#)

## **Utilisation des événements du consignateur**



Utilisez cette page pour savoir comment utiliser les événements du consignateur afin de déterminer les extensions de journal qui ne sont plus requises pour le redémarrage du gestionnaire de files d'attente ou la reprise sur incident lié au support.

Vous pouvez archiver les extensions de journal superflues sur un support tel qu'une bande pour la reprise après incident avant de les supprimer du répertoire des journaux actifs. La suppression régulière des extensions de journal superflues réduit au minimum l'utilisation de l'espace disque.

Si l'attribut de gestionnaire de files d'attente LOGGERS est activé, mais qu'un message d'événement du consignateur ne peut pas être inséré dans la file d'attente d'événements du consignateur, par exemple parce que la file d'attente d'événements n'a pas été définie, l'action à l'origine de l'événement se poursuit.

### Concepts associés

«Événements de journalisation», à la page 51

Les événements du consignateur sont des notifications indiquant qu'un gestionnaire de files d'attente a commencé à écrire dans un nouveau domaine réservé aux journaux  ou, sous IBM i, dans un récepteur de journal.  Les messages d'événement du consignateur ne sont pas disponibles avec IBM MQ for z/OS.

### Référence associée

[LoggerEvent \(MQLONG\)](#)

[LoggerEvent \(entier signé à 10 chiffres\)](#)

«Génération d'événements de consignateur», à la page 51

Cette page permet d'afficher les situations qui entraînent la génération d'événements de consignateur et de comprendre les circonstances dans lesquelles les événements de consignateur ne sont pas générés.

## **Multi Exemple de programme C permettant de surveiller la file d'attente d'événements du consignateur**

Cette page permet d'afficher un exemple de programme C qui surveille la file d'attente d'événements du consignateur pour détecter de nouveaux messages d'événement, lit ces messages et place le contenu du message dans stdout.

```
/*
/*
/* Program name: AMQSLOG0.C
/*
/*
/* Description: Sample C program to monitor the logger event queue and
/* display formatted message content to stdout when a logger
/* event occurs
/*
/* <copyright
/* notice="lm-source-program"
/* pids="5724-H72,"
/* years="2005, 2024"
/* crc="186943832" >
/* Licensed Materials - Property of IBM
/*
/*
/* 5724-H72,
/*
/* (C) Copyright IBM Corp. 2005, 2024. All Rights Reserved.
/*
/* US Government Users Restricted Rights - Use, duplication or
/* disclosure restricted by GSA ADP Schedule Contract with
/* IBM Corp.
/* </copyright>
/*
/*
/* Function: AMQSLOG is a sample program which monitors the logger event
/* queue for new event messages, reads those messages, and displays the
/* formatted contents of the message to stdout.
/*
/*
/*
/* AMQSLOG has 1 parameter - the queue manager name (optional, if not
/* specified then the default queue manager is implied)
/*
/*
/*
/* Includes
/*
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

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

/* Constants
/*
/*
#define MAX_MESSAGE_LENGTH 8000

typedef struct _ParmTableEntry
{
    MQLONG ConstVal;
    PMQCHAR Desc;
} ParmTableEntry;

ParmTableEntry ParmTable[] =
{
    {0, ""},
    {MQCA_Q_MGR_NAME, "Queue Manager Name"},
    {MQCMD_LOGGER_EVENT, "Logger Event Command"},
    {MQRC_LOGGER_STATUS, "Logger Status"},
    {MQCACF_ARCHIVE_LOG_EXTENT_NAME, "Archive Log Extent"},
    {MQCACF_CURRENT_LOG_EXTENT_NAME, "Current Log Extent"},
    {MQCACF_RESTART_LOG_EXTENT_NAME, "Restart Log Extent"},
}
```

```

    {MQCACF_MEDIA_LOG_EXTENT_NAME , "Media Log Extent"},
    {MQCACF_LOG_PATH              , "Log Path"}
};

#if defined(MQ_64_BIT)
    #define Int32
#else
    #define Int32 "l"
#endif

/*****
/* Function prototypes
*****/

static void ProcessPCF(MQHCONN      hConn,
                      MQHOBJ      hEventQueue,
                      PMQCHAR      pBuffer);

static PMQCHAR ParmToString(MQLONG Parameter);

/*****
/* Function: main
*****/

int main(int argc, char * argv[])
{
    MQLONG      CompCode;
    MQLONG      Reason;
    MQHCONN     hConn = MQHC_UNUSABLE_HCONN;
    MQOD        ObjDesc = { MQOD_DEFAULT };
    MQCHAR      QMName[MQ_Q_MGR_NAME_LENGTH+1] = "";
    MQCHAR      LogEvQ[MQ_Q_NAME_LENGTH] = "SYSTEM.ADMIN.LOGGER.EVENT";
    MQHOBJ      hEventQueue = MQHO_UNUSABLE_HOBJ;
    PMQCHAR     pBuffer = NULL;

    printf("\n/*****/\n");
    printf("/* Sample Logger Event Monitor start */\n");
    printf("/*****/\n");

    /*****
    /* Parse any command line options
    *****/
    if (argc > 1)
    {
        strncpy(QMName, argv[1], (size_t)MQ_Q_MGR_NAME_LENGTH);
    }

    pBuffer = (PMQCHAR)malloc(MAX_MESSAGE_LENGTH);
    if (pBuffer == NULL)
    {
        printf("Can't allocate %d bytes\n", MAX_MESSAGE_LENGTH);
        goto MOD_EXIT;
    }

    /*****
    /* Connect to the specified (or default) queue manager
    *****/
    MQCONN( QMName,
            &hConn,
            &CompCode,
            &Reason);

    if (Reason != MQRC_NONE)
    {
        printf("MQCONN ended with reason code %" Int32 "d\n", Reason);
        goto MOD_EXIT;
    }

    /*****
    /* Open the logger event queue for input
    *****/
    strncpy(ObjDesc.ObjectQMgrName, QMName, MQ_Q_MGR_NAME_LENGTH);
    strncpy(ObjDesc.ObjectName, LogEvQ, MQ_Q_NAME_LENGTH);

    MQOPEN( hConn,
            &ObjDesc,
            MQOO_INPUT_EXCLUSIVE,
            &hEventQueue,
            &CompCode,
            &Reason );

    if (Reason != MQRC_NONE)

```

```

    {
        printf("MQOPEN failed for queue manager %.48s Queue %.48s Reason: %" Int32 "d\n",
              ObjDesc.ObjectQMgrName,
              ObjDesc.ObjectName,
              Reason);
        goto MOD_EXIT;
    }
else
{
    /* Start processing event messages */
    ProcessPCF(hConn, hEventQueue, pBuffer);
}

MOD_EXIT:
if (pBuffer != NULL)
{
    free(pBuffer);
}

/* Close the logger event queue */
if (hEventQueue != MQHO_UNUSABLE_HOBJ)
{
    MQCLOSE(hConn, &hEventQueue, MQCO_NONE, &CompCode, &Reason);
}

/* Disconnect */
if (hConn != MQHC_UNUSABLE_HCONN)
{
    MQDISC(&hConn, &CompCode, &Reason);
}

return 0;
}

/* Function: ProcessPCF */
/* Input Parameters: Handle to queue manager connection */
/*                   Handle to the opened logger event queue object */
/*                   Pointer to a memory buffer to store the incoming PCF */
/*                   message */
/* Output Parameters: None */
/* Logic: Wait for messages to appear on the logger event queue and display */
/*        their formatted contents. */
/* ***** */

static void ProcessPCF(MQHCONN hConn,
                     MQHOBJ hEventQueue,
                     PMQCHAR pBuffer)
{
    MQCFH * pCfh;
    MQCFST * pCfst;
    MQGMO Gmo = { MQGMO_DEFAULT };
    MQMD Mqmd = { MQMD_DEFAULT };
    PMQCHAR pPCFCmd;
    MQLONG CompCode = MQCC_OK;
    MQLONG Reason = MQRC_NONE;
    MQLONG MsgLen;
    PMQCHAR Parm = NULL;

    Gmo.Options = MQGMO_WAIT +
                 MQGMO_CONVERT +
                 MQGMO_FAIL_IF QUIESCING;
    Gmo.WaitInterval = MQWI_UNLIMITED; /* Set timeout value */

    /* Process response Queue */
    while (Reason == MQRC_NONE)
    {
        memcpy(&Mqmd.MsgId, MQMI_NONE, sizeof(Mqmd.MsgId));
    }
}

```

```

memset(&Mqmd.CorrelId, 0, sizeof(Mqmd.CorrelId));

MQGET( hConn,
       hEventQueue,
       &Mqmd,
       &Gmo,
       MAX_MESSAGE_LENGTH,
       pBuffer,
       &MsgLen,
       &CompCode,
       &Reason );

if (Reason != MQRC_NONE)
{
    switch(Reason)
    {
        case MQRC_NO_MSG_AVAILABLE:
            printf("Timed out");
            break;

        default:
            printf("MQGET ended with reason code %" Int32 "d\n", Reason);
            break;
    }
    goto MOD_EXIT;
}

/*****
/* Only expect PCF event messages on this queue */
/*****
if (memcmp(Mqmd.Format, MQFMT_EVENT, MQ_FORMAT_LENGTH))
{
    printf("Unexpected message format '%8.8s' received\n", Mqmd.Format);
    continue;
}

/*****
/* Build the output by parsing the received PCF message, first the */
/* header, then each of the parameters */
/*****
pCfh = (MQCFH *)pBuffer;

if (pCfh->Reason != MQRC_NONE)
{
    printf("-----\n");
    printf("Event Message Received\n");

    Parm = ParmToString(pCfh->Command);
    if (Parm != NULL)
    {
        printf("Command  :%s \n",Parm);
    }
    else
    {
        printf("Command  :%" Int32 "d \n",pCfh->Command);
    }

    printf("CompCode :%" Int32 "d\n" ,pCfh->CompCode);

    Parm = ParmToString(pCfh->Reason);
    if (Parm != NULL)
    {
        printf("Reason   :%s \n",Parm);
    }
    else
    {
        printf("Reason   :%" Int32 "d \n",pCfh->Reason);
    }
}

pPCFCmd = (PMQCHAR) (pCfh+1);
printf("-----\n");
while(pCfh->ParameterCount-->0)
{
    pCfst = (MQCFST *) pPCFCmd;
    switch(pCfst->Type)
    {
        case MQCFT_STRING:
            Parm = ParmToString(pCfst->Parameter);
            if (Parm != NULL)
            {
                printf("%-32s",Parm);
            }
        }
    }
}

```



```

    }
    else
    {
        printf("%-32s Int32 %d", pCfst->Parameter);
    }

    fwrite(pCfst->String, pCfst->StringLength, 1, stdout);
    pPCFCmd += pCfst->StrucLength;
    break;

default:
    printf("Unrecognised datatype %s Int32 %d returned\n", pCfst->Type);
    goto MOD_EXIT;
}
}
putchar('\n');
}

```

## Exemple de sortie

Cette application génère la forme de sortie suivante:

```

/*****
/* Sample Logger Event Monitor start */
*****/
-----
Event Message Received
Command :Logger Event Command
CompCode :0
Reason :Logger Status
-----
Queue Manager Name          CSIM

Current Log Extent          AMQA000001
Restart Log Extent         AMQA000001
Media Log Extent           AMQA000001
Log Path                    QMCSIM
-----

```

### Concepts associés

[«Utilisation des événements du consignateur», à la page 52](#)

Utilisez cette page pour savoir comment utiliser les événements du consignateur afin de déterminer les extensions de journal qui ne sont plus requises pour le redémarrage du gestionnaire de files d'attente ou la reprise sur incident lié au support.

[«Utilisation de l'événement de commande», à la page 50](#)

Utilisez cette page pour voir comment vous pouvez utiliser les événements de commande pour générer une trace d'audit des commandes qui ont été exécutées

### Référence associée

[«Génération d'événements de consignateur», à la page 51](#)

Cette page permet d'afficher les situations qui entraînent la génération d'événements de consignateur et de comprendre les circonstances dans lesquelles les événements de consignateur ne sont pas générés.

## Événements de configuration de droits d'accès

Les événements de configuration des droits d'accès sont générés lorsqu'une modification est apportée à partir de l'une des opérations de contrôle de sécurité via la ligne de commande, MQSC, PCF ou les commandes IBM i correspondantes.

Les données d'événement contiennent les informations suivantes:

### Informations d'origine

comprend le gestionnaire de files d'attente à partir duquel la modification a été effectuée, l'ID de l'utilisateur qui a effectué la modification et la manière dont la modification a été apportée, par exemple par une commande de console.

### Informations contextuelles

une réplique des informations de contexte dans les données de message du message de commande.

Les informations contextuelles sont incluses dans les données d'événement lorsque la commande a été entrée en tant que message dans SYSTEM.ADMIN.COMMAND.QUEUE .

### **Identité d'enregistrement de droits d'accès**

comprend le nom de profil et le type d'objet de l'enregistrement de droits d'accès.

### **Attributs d'objet**

comprend les valeurs de tous les attributs de l'enregistrement de droits d'accès.

Dans le cas d'événements d'enregistrement de droits de modification, deux messages sont générés, l'un avec les informations avant la modification, l'autre avec les informations après la modification.

Chaque message d'événement généré est placé dans SYSTEM.ADMIN.CONFIG.EVENT EVENT.

### **Référence associée**

«Types d'événements», à la page 12

Cette page permet d'afficher les types d'événement d'instrumentation qu'un gestionnaire de files d'attente ou une instance de canal peut signaler.

### **Génération d'événements de configuration des droits d'accès**

Cette page permet d'afficher les situations qui génèrent des événements de configuration des droits d'accès et de comprendre les circonstances dans lesquelles les événements de configuration des droits d'accès ne sont pas générés.

Les événements de configuration des droits vous informent des modifications apportées aux attributs d'un enregistrement de droits d'accès. Il existe trois types d'événement de configuration des droits:

- [Modifier l'enregistrement de droits d'accès](#)
- [Supprimer l'enregistrement de droits d'accès](#)
- [Actualiser l'enregistrement de droits d'accès](#)

Un message d'événement d'autorité est inséré dans la file d'attente d'événements de configuration, lorsque l'attribut de gestionnaire de files d'attente **CONFIGEV** est défini sur *ENABLED* et que l'une des commandes suivantes, ou leur équivalent MQSC, est émis, même s'il n'y a pas de modification réelle de l'enregistrement d'autorité:

- Commande PCF [Delete Authority Record](#)
- Commande PCF [Set Authority Record](#)
- Commande de contrôle [setmqaut](#)
- Commande CL [RVKMQMAUT](#)
- Commande CL [GRMQMAUT](#)

### **Lorsque les événements de configuration des droits ne sont pas générés**

Les messages d'événement de configuration des droits d'accès ne sont pas générés dans les cas suivants:

- En cas d'échec d'une commande
- Lorsqu'un gestionnaire de files d'attente rencontre une erreur lors de la tentative d'insertion d'un message dans la file d'attente d'événements, auquel cas la commande se termine, mais aucun message d'événement n'est généré
- Lors de la création ou de la suppression d'un gestionnaire de files d'attente
- Lorsqu'un objet est supprimé, quelle que soit l'option **AUTHREC** de la commande de suppression. L'événement de commande correspondant indique cette opération, qui ne s'applique pas à l'enregistrement de droits d'accès pour les utilisateurs individuels.

### **Concepts associés**

«Événements Commande», à la page 49

Les événements de commande sont des notifications indiquant qu'une commande MQSC ou PCF a été exécutée avec succès.

## Référence associée

ACTUALISEZ LE GESTIONNAIRE DE FILES D'ATTENTE

## Multi Exemple de programme de surveillance des événements d'instrumentation sur Multiplatforms

**amqsevt** formate les événements d'instrumentation qu'un gestionnaire de files d'attente peut créer et est fourni avec IBM MQ for Multiplatforms. Le programme lit les messages des files d'attente d'événements et les convertit en chaînes lisibles.

En tant qu'exemple de programme, source et binaire sont fournis. L'exemple est fourni sur tous les Multiplatforms, y compris IBM i.

Le fichier binaire unique `amqsevt` (ou `amqsevt.exe`) est fourni dans l'ensemble de fichiers des exemples et est installé dans le répertoire `bin` des exemples (`tools\c\samples\bin` ou `bin64`).

Les fichiers source `amqsevt.c` sont également fournis dans l'ensemble de fichiers des exemples et sont installés dans le répertoire des exemples, c'est-à-dire `tools\c\samples` sur Windows.

Notez que le programme peut lire plusieurs files d'attente d'événements et s'abonner à plusieurs rubriques à l'aide de `MQCB` pour extraire les messages.

Lorsqu'il est exécuté en tant que client, l'exemple peut se connecter à n'importe quel gestionnaire de files d'attente, y compris z/OS.

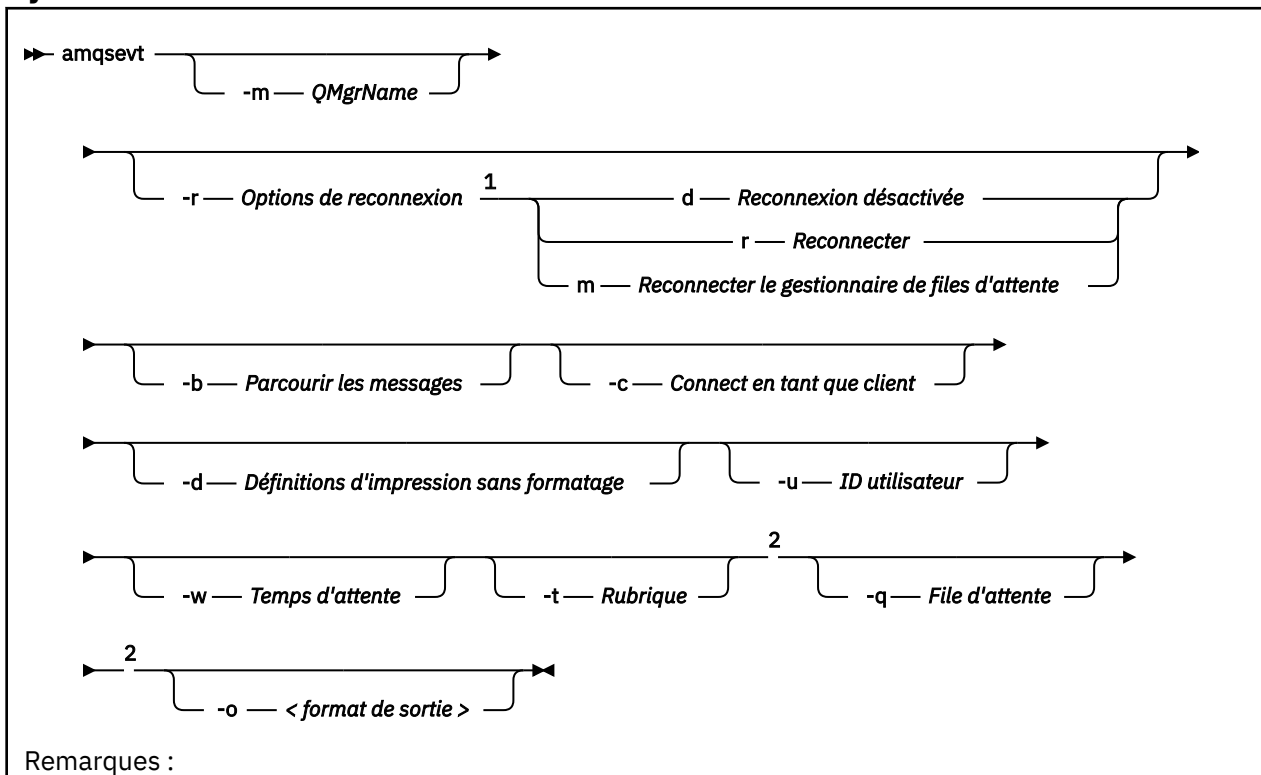


**Avvertissement :** Vous pouvez utiliser le programme sans spécifier de paramètres, auquel cas le programme tente de se connecter au gestionnaire de files d'attente par défaut et de lire les messages de l'ensemble standard de files d'attente d'événements (`SYSTEM.ADMIN.*.ÉVÉNEMENT`).

Dans ce cas, le programme attend indéfiniment les messages jusqu'à ce que vous appuyez sur la touche Entrée pour mettre fin au programme.

Cependant, vous êtes plus susceptible d'utiliser le programme avec les différentes options décrites.

## Syntaxe



<sup>1</sup> Disponible uniquement en cas d'utilisation en tant que client

<sup>2</sup> Les files d'attente et les rubriques peuvent comporter plusieurs entrées

## Paramètres facultatifs

### **-m *QueueManagerName***

Indiquez un gestionnaire de files d'attente spécifique pour la lecture des événements.

### **-r *Reconnection Options***

Options de reconnexion automatique lorsqu'elles sont utilisées en tant que client. Les valeurs possibles sont les suivantes:

**d**

Reconnexion du client désactivée

**r**

Reconnecter le client

**m**

Reconnexion du gestionnaire de files d'attente

### **-b**

Parcourir uniquement les enregistrements, plutôt que de lire les messages de façon destructive

### **-c**

Sélectionne la connexion en tant que client.

### **-d**

Sélectionne le mode d'impression utilisé dans le deuxième exemple. Les constantes MQI sont imprimées exactement telles qu'elles apparaissent dans les fichiers d'en-tête.

### **-u *User ID***

Indique un utilisateur spécifique et fait apparaître une invite demandant un mot de passe

### **-w *Wait***

Provoque la sortie du programme si aucun message d'événement n'est arrivé dans le nombre de secondes indiqué.

Notez que si vous n'indiquez pas d'heure, le programme ne s'arrête normalement que lorsque vous appuyez sur la touche ENTREE.

### **-t *Topic* et**

### **-q *Queue***

Les options **-q** et **-t** peuvent être fournies plusieurs fois sur la ligne de commande.

Par conséquent, il est possible de lire à partir de certaines files d'attente standard et également à partir de rubriques (si des événements leur sont envoyés) à partir d'une seule exécution du programme.

Si aucune file d'attente ou rubrique n'est indiquée sur la ligne de commande, les files d'attente d'événements par défaut sont ouvertes.

**Remarque :** Le programme détecte s'il s'est connecté à un gestionnaire de files d'attente z/OS en tant que client et modifie l'ensemble par défaut des files d'attente d'événements de manière appropriée, car z/OS ne dispose pas de SYSTEM.ADMIN.LOGGER.EVENT EVENT.

Lorsque des rubriques sont utilisées, le programme utilise un abonnement non durable avec une file d'attente gérée afin que tout soit nettoyé lorsqu'il se ferme.

### **-o <output format>**

Format de la sortie. Les valeurs possibles sont les suivantes :

**texte**

Format de texte standard ; il s'agit de la valeur par défaut.

## json

Format JSON standard ; toute application qui comprend JSON peut prendre cette sortie et la traiter directement.

## Exemple de sortie

Les trois exemples suivants illustrent la sortie du programme.

Le premier exemple utilise l'option de formatage par défaut dans laquelle le programme utilise la définition MQI d'une zone et formate la sortie pour la rendre plus lisible.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : Queue Mgr Event
Reason               : Unknown Alias Base Queue
Event created        : 2015/06/17 13:47:07.02 GMT
  Queue Mgr Name     : V8003_A
  Queue Name         : EVT.NO.BASE.QUEUE
  Base Object Name   : EVT.NOT.DEFINED
  Appl Type          : Unix
  Appl Name          : amqsput
  Base Type          : Queue
```

Le deuxième exemple illustre le formatage alternatif, à l'aide de l'option -d , qui ne tente pas de traduire les constantes MQI. Cela peut être préférable pour certains outils de script qui recherchent des valeurs MQI spécifiques.

```
**** Message (320 Bytes) on Queue SYSTEM.ADMIN.QMGR.EVENT ****
Event Type           : MQCMD_Q_MGR_EVENT
Reason               : MQRC_UNKNOWN_ALIAS_BASE_Q
Event created        : 2015/06/17 13:52:48.18 GMT
  MQCA_Q_MGR_NAME    : V8003_A
  MQCA_Q_NAME        : EVT.NO.BASE.QUEUE
  MQCA_BASE_OBJECT_NAME : EVT.NOT.DEFINED
  MQIA_APPL_TYPE     : MQAT_UNIX
  MQCACF_APPL_NAME   : amqsput
  MQIA_BASE_TYPE     : MQOT_Q
```

Le troisième exemple illustre la sortie JSON de la commande

```
amqsevt -m QM1 -q SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE -o json
```

```
{
  "eventSource" : { "objectName": "SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE",
                   "objectType" : "Queue" },
  "eventType" : {
    "name" : "Activity Trace",
    "value" : 209
  },
  "eventReason" : {
    "name" : "None",
    "value" : 0
  },
  "eventCreation" : {
    "timeStamp" : "2018-07-10T12:44:26Z",
    "epoch" : 1531226666
  },
  "eventData" : {
    "queueMgrName" : "QM1",
    "hostName" : "<yourhostname>",
    "startDate" : "2018-07-10",
    "startTime" : "13:44:25",
    "endDate" : "2018-07-10",
    "endTime" : "13:44:26",
    "commandLevel" : 910,
    ...
  }
}
```

## Utilisation d'exemple

L'exemple suivant montre comment utiliser plusieurs files d'attente:

```
amqsevt -m QM1 -q SYSTEM.ADMIN.QMGR.EVENT -q SYSTEM.ADMIN.PERM.EVENT -w 1
```

### Concepts associés

[«Moniteur d'événements», à la page 9](#)

La surveillance des événements consiste à détecter les occurrences d' *événements d'instrumentation* dans un réseau de gestionnaires de files d'attente. Un événement d'instrumentation est une combinaison logique d'événements détectés par un gestionnaire de files d'attente ou une instance de canal. Un tel événement entraîne le gestionnaire de files d'attente ou l'instance de canal à insérer un message spécial, appelé *message d'événement*, dans une file d'attente d'événements.

[«Événements d'instrumentation», à la page 10](#)

Un événement d'instrumentation est une combinaison logique de conditions qu'un gestionnaire de files d'attente ou une instance de canal détecte et insère un message spécial, appelé *message d'événement*, dans une file d'attente d'événements.

### Référence associée

[Programmation C](#)

[«Exemple de programme C permettant de surveiller la file d'attente d'événements du consignateur», à la page 53](#)

Cette page permet d'afficher un exemple de programme C qui surveille la file d'attente d'événements du consignateur pour détecter de nouveaux messages d'événement, lit ces messages et place le contenu du message dans stdout.

## Surveillance de message

La surveillance des messages consiste à identifier la route empruntée par un message via un réseau de gestionnaires de files d'attente. En identifiant les types d'activités et la séquence d'activités effectuées pour le compte d'un message, vous pouvez déterminer la route du message.

Au fur et à mesure qu'un message passe par un réseau de gestionnaires de files d'attente, divers processus effectuent des activités pour le compte du message. Utilisez l'une des techniques suivantes pour déterminer une route de message:

- L'application d'affichage de route IBM MQ (dspmqrte)
- Enregistrement de l'activité
- Messagerie de routage de trace

Ces techniques génèrent toutes des messages spéciaux qui contiennent des informations sur les activités effectuées sur le message lors de sa transmission via un réseau de gestionnaires de files d'attente.

Utilisez les informations renvoyées dans ces messages spéciaux pour atteindre les objectifs suivants:

- Enregistrez l'activité des messages.
- Déterminez le dernier emplacement connu d'un message.
- Détectez les problèmes de routage dans votre réseau de gestionnaires de files d'attente.
- Aide à déterminer les causes des problèmes de routage dans votre réseau de gestionnaires de files d'attente.
- Vérifiez que votre réseau de gestionnaires de files d'attente s'exécute correctement.
- Familiarisez-vous avec l'exécution de votre réseau de gestionnaires de files d'attente.
- Trace des messages publiés.

### Concepts associés

[Types de message](#)

## Activités et opérations

Les activités sont des actions discrètes qu'une application effectue pour le compte d'un message. Les activités consistent en des opérations, qui sont des éléments de travail uniques effectués par une application.

Les actions suivantes sont des exemples d'activités:

- Un agent MCA envoie un message à partir d'une file d'attente de transmission vers un canal
- Un agent MCA reçoit un message d'un canal et le place dans sa file d'attente cible
- Une application extrait un message d'une file d'attente et place un message de réponse dans la réponse.
- Le moteur de publication / abonnement IBM MQ traite un message.

Les activités consistent en une ou plusieurs *opérations*. Les opérations sont des éléments de travail uniques qu'une application effectue. Par exemple, l'activité d'un agent MCA envoyant un message à partir d'une file d'attente de transmission vers le bas d'un canal comprend les opérations suivantes:

1. Obtention d'un message à partir d'une file d'attente de transmission (opération *Obtenir*).
2. Envoi du message via un canal (opération *Envoyer*).

Dans un réseau de publication / abonnement, l'activité du moteur de publication / abonnement IBM MQ traitant un message peut se composer des opérations multiples suivantes:

1. Insertion d'un message dans une chaîne de rubrique (opération *Put*).
2. Zéro ou plusieurs opérations pour chacun des abonnés pris en compte pour la réception du message (une opération *Publier*, une opération *Publier supprimé* ou une opération *Publier exclue*).

## Informations provenant des activités

Vous pouvez identifier la séquence des activités effectuées sur un message en enregistrant les informations lorsque le message est acheminé via un réseau de gestionnaires de files d'attente. Vous pouvez déterminer la route d'un message via le réseau du gestionnaire de files d'attente à partir de la séquence des activités effectuées sur le message et obtenir les informations suivantes:

### Dernier emplacement connu d'un message

Si un message n'atteint pas sa destination prévue, vous pouvez déterminer le dernier emplacement connu du message à partir d'une route de message complète ou partielle.

### Problèmes de configuration avec un réseau de gestionnaires de files d'attente

Lorsque vous étudiez la route d'un message via un réseau de gestionnaires de files d'attente, vous pouvez constater que le message n'est pas allé à l'endroit prévu. Il existe de nombreuses raisons pour lesquelles cela peut se produire, par exemple, si un canal est inactif, le message peut prendre une autre route.

Pour une application de publication / abonnement, vous pouvez également déterminer la route d'un message en cours de publication vers une rubrique et tous les messages qui circulent dans un réseau de gestionnaires de files d'attente suite à la publication aux abonnés.

Dans de telles situations, un administrateur système peut déterminer s'il existe des problèmes dans le réseau du gestionnaire de files d'attente et, le cas échéant, les corriger.

## Routes des messages

En fonction de la raison pour laquelle vous avez déterminé une route de message, vous pouvez utiliser les approches générales suivantes:

### Utilisation des informations d'activité enregistrées pour un message de suivi

Les messages de trace-route enregistrent les informations d'activité dans un but spécifique. Vous pouvez les utiliser pour déterminer les problèmes de configuration d'un réseau de gestionnaires de files d'attente ou pour déterminer le dernier emplacement connu d'un message. Si un message de trace-route est généré pour déterminer le dernier emplacement connu d'un message qui n'a pas

atteint sa destination prévue, il peut imiter le message d'origine. Cela donne au message trace-route la plus grande chance de suivre la route empruntée par le message d'origine.

L'application d'affichage de route IBM MQ peut générer des messages de trace.

### **Utilisation des informations d'activité enregistrées pour le message d'origine**

Vous pouvez activer n'importe quel message pour l'enregistrement d'activité et faire en sorte que les informations d'activité soient enregistrées en son nom. Si un message n'atteint pas sa destination prévue, vous pouvez utiliser les informations d'activité enregistrées pour déterminer le dernier emplacement connu du message. En utilisant les informations d'activité du message d'origine, la route de message la plus précise possible peut être déterminée, conduisant au dernier emplacement connu. Pour utiliser cette approche, le message d'origine doit être activé pour l'enregistrement d'activité.

**Avertissement :** Evitez d'activer tous les messages d'un réseau de gestionnaires de files d'attente pour l'enregistrement de l'activité. Les messages activés pour l'enregistrement d'activité peuvent avoir de nombreux rapports d'activité générés en leur nom. Si chaque message d'un réseau de gestionnaires de files d'attente est activé pour l'enregistrement de l'activité, le trafic réseau du gestionnaire de files d'attente peut augmenter jusqu'à un niveau inacceptable.

### **Concepts associés**

«Surveillance de message», à la page 62

La surveillance des messages consiste à identifier la route empruntée par un message via un réseau de gestionnaires de files d'attente. En identifiant les types d'activités et la séquence d'activités effectuées pour le compte d'un message, vous pouvez déterminer la route du message.

«Techniques de routage des messages», à la page 64

L'enregistrement d'activité et la messagerie de suivi sont des techniques qui permettent d'enregistrer les informations d'activité d'un message lors de son acheminement via un réseau de gestionnaires de files d'attente.

«Messagerie de routage de trace», à la page 71

La messagerie d'acheminement de trace est une technique qui utilise des *messages d'acheminement de trace* pour enregistrer les informations d'activité d'un message. La messagerie d'acheminement de trace implique l'envoi d'un message d'acheminement de trace dans un réseau de gestionnaires de files d'attente.

### **Tâches associées**

Écriture de vos propres agents MCA

## **Techniques de routage des messages**

L'enregistrement d'activité et la messagerie de suivi sont des techniques qui permettent d'enregistrer les informations d'activité d'un message lors de son acheminement via un réseau de gestionnaires de files d'attente.

### **Enregistrement de l'activité**

Si un message a l'option de rapport appropriée spécifiée, il demande que les applications génèrent des *rapports d'activité* lorsqu'il est acheminé via un réseau de gestionnaires de files d'attente.

Lorsqu'une application effectue une activité pour le compte d'un message, un rapport d'activité peut être généré et distribué à un emplacement approprié. Un rapport d'activité contient des informations sur l'activité effectuée sur le message.

Les informations d'activité collectées à l'aide des rapports d'activité doivent être organisées dans l'ordre avant qu'une route de message puisse être déterminée.

### **Messagerie de routage de trace**

La *messagerie de trace-route* est une technique qui consiste à envoyer un *message de trace-route* dans un réseau de gestionnaires de files d'attente. Lorsqu'une application effectue une activité pour le compte du message de suivi, les informations d'activité peuvent être accumulées dans les données de message du message de suivi ou des rapports d'activité peuvent être générés. Si des informations d'activité sont accumulées dans les données de message du message de routage de trace, lorsqu'il



atteint sa file d'attente cible, un message de réponse de routage de trace contenant toutes les informations du message de routage de trace peut être généré et distribué à un emplacement approprié.

Etant donné qu'un message de suivi est dédié à l'enregistrement de la séquence des activités effectuées en son nom, il existe davantage d'options de traitement que les messages normaux qui demandent des rapports d'activité.

### Comparaison de l'enregistrement d'activité et de la messagerie de suivi

L'enregistrement d'activité et la messagerie de routage peuvent fournir des informations d'activité pour déterminer la route empruntée par un message via un réseau de gestionnaires de files d'attente. Les deux méthodes ont leurs propres avantages.

Bénéfice	Enregistrement de l'activité	Messagerie de routage de trace
Peut déterminer le dernier emplacement connu d'un message	Oui	Oui
Peut déterminer les problèmes de configuration avec un réseau de gestionnaires de files d'attente	Oui	Oui
Peut être demandé par n'importe quel message (n'est pas limité à l'utilisation avec les messages de trace-route)	Oui	Non
Les données de message ne sont pas modifiées	Oui	Non
Message traité normalement	Oui	Non
Les informations d'activité peuvent être accumulées dans les données de message	Non	Oui
Distribution facultative des messages à la file d'attente cible	Non	Oui
Si un message est intercepté dans une boucle sans fin, il peut être détecté et traité	Non	Oui
Les informations sur les activités peuvent être mises en ordre de manière fiable	Non	Oui
Application fournie pour afficher les informations d'activité	Non	Oui

### Exhaustivité de la route des messages

Dans certains cas, il n'est pas possible d'identifier la séquence complète des activités effectuées pour le compte d'un message, de sorte que seule une route de message partielle peut être déterminée. L'exhaustivité d'une route de message est directement influencée par le réseau de gestionnaires de files d'attente via lequel les messages sont acheminés.

Les agents MCA et les applications écrites par l'utilisateur connectés aux gestionnaires de files d'attente peuvent enregistrer des informations relatives aux activités effectuées pour le compte d'un message. L'enregistrement des informations d'activité est contrôlé par les attributs de gestionnaire de files d'attente ACTIVREC et ROUTEREC. Un réseau de gestionnaires de files d'attente peut déterminer les routes de messages complètes.

### Mode de stockage des informations d'activité

IBM MQ stocke les informations d'activité dans des rapports d'activité, des messages de trace-route ou des messages de réponse de trace-route. Dans chaque cas, les informations sont stockées dans une structure appelée groupe PCF *Activité*. Un message de suivi ou un message de réponse de suivi peut contenir de nombreux groupes PCF d'activité, en fonction du nombre d'activités effectuées sur le

message. Les rapports d'activité contiennent un groupe PCF d'activité car un rapport d'activité distinct est généré pour chaque activité enregistrée.

Avec la messagerie de trace-route, des informations supplémentaires peuvent être enregistrées. Ces informations supplémentaires sont stockées dans une structure appelée groupe PCF *TraceRoute*. Le groupe PCF *TraceRoute* contient un certain nombre de structures PCF qui sont utilisées pour stocker des informations d'activité supplémentaires et pour spécifier des options qui déterminent la façon dont le message de routage de trace est traité lorsqu'il est acheminé via un réseau de gestionnaires de files d'attente.

### Concepts associés

«Enregistrement de l'activité», à la page 66

L'enregistrement d'activité est une technique permettant de déterminer les routes que les messages empruntent via un réseau de gestionnaires de files d'attente. Pour déterminer la route empruntée par un message, les activités effectuées pour le compte du message sont enregistrées.

«Messagerie de routage de trace», à la page 71

La messagerie d'acheminement de trace est une technique qui utilise des *messages d'acheminement de trace* pour enregistrer les informations d'activité d'un message. La messagerie d'acheminement de trace implique l'envoi d'un message d'acheminement de trace dans un réseau de gestionnaires de files d'attente.

### Référence associée

«Groupe PCF *TraceRoute*», à la page 77

Les attributs du groupe PCF *TraceRoute* contrôlent le comportement d'un message de suivi. Le groupe PCF *TraceRoute* se trouve dans les données de message de chaque message de suivi.

«Données de message du rapport d'activité», à la page 113

Cette page permet d'afficher les paramètres contenus par le groupe PCF *Activité* dans un message de rapport d'activité. Certains paramètres sont renvoyés uniquement lorsque des opérations spécifiques ont été effectuées.

## Enregistrement de l'activité

L'enregistrement d'activité est une technique permettant de déterminer les routes que les messages empruntent via un réseau de gestionnaires de files d'attente. Pour déterminer la route empruntée par un message, les activités effectuées pour le compte du message sont enregistrées.

Lors de l'utilisation de l'enregistrement d'activité, chaque activité effectuée pour le compte d'un message peut être enregistrée dans un rapport d'activité. Un rapport d'activités est un type de message de rapport. Chaque rapport d'activité contient des informations sur l'application qui a effectué l'activité pour le compte du message, la date à laquelle l'activité a eu lieu et des informations sur les opérations effectuées dans le cadre de l'activité. Les rapports d'activité sont généralement distribués dans une file d'attente de réponse où ils sont collectés ensemble. En étudiant les rapports d'activité liés à un message, vous pouvez déterminer la route empruntée par le message via le réseau du gestionnaire de files d'attente.

## Utilisation du rapport d'activité

Lorsque des messages sont acheminés via un réseau de gestionnaires de files d'attente, des rapports d'activité peuvent être générés. Vous pouvez utiliser les informations du rapport d'activités de l'une des manières suivantes:

### Déterminer le dernier emplacement connu d'un message

Si un message activé pour l'enregistrement d'activité n'atteint pas sa destination prévue, les rapports d'activité générés pour le message lors de son acheminement via un réseau de gestionnaires de files d'attente peuvent être étudiés afin de déterminer le dernier emplacement connu du message.

### Détermination des problèmes de configuration liés à un réseau de gestionnaires de files d'attente

Un certain nombre de messages activés pour l'enregistrement d'activité peuvent être envoyés dans un réseau de gestionnaires de files d'attente. En étudiant les rapports d'activité relatifs à chaque message, il peut apparaître qu'ils n'ont pas pris la route prévue. Il existe de nombreuses raisons pour lesquelles cela peut se produire, par exemple, un canal peut s'être arrêté, forçant le message à

prendre une autre route. Dans ces situations, un administrateur système peut déterminer s'il y a des problèmes dans le réseau du gestionnaire de files d'attente et, le cas échéant, les corriger.

**Remarque :** Vous pouvez utiliser l'enregistrement d'activité avec les messages de trace-route à l'aide de l'application d'affichage de route IBM MQ .

## Format du rapport d'activité

Les rapports d'activité sont des messages PCF générés par des applications qui ont effectué une activité pour le compte d'un message. Les rapports d'activité sont des messages de rapport IBM MQ standard contenant un descripteur de message et des données de message, comme suit:

### Descripteur de message

- Une structure MQMD

### Données de message

- Un en-tête PCF intégré (MQEPH)
- Données de message du rapport d'activité

Les données de message du rapport d'activité sont constituées du groupe PCF *Activity* et, si elles sont générées pour un message de suivi, du groupe PCF *TraceRoute* .

### Référence associée

[MQMD-Descripteur de message](#)

[MQEPH-en-tête PCF imbriqué](#)

## Contrôle de l'enregistrement des activités

Activez l'enregistrement des activités au niveau du gestionnaire de files d'attente. Pour activer l'intégralité d'un réseau de gestionnaires de files d'attente, activez individuellement chaque gestionnaire de files d'attente du réseau pour l'enregistrement de l'activité. Si vous activez davantage de gestionnaires de files d'attente, des rapports d'activité supplémentaires sont générés.

## Pourquoi et quand exécuter cette tâche

Pour générer des rapports d'activité pour un message lors de son acheminement via un gestionnaire de files d'attente, définissez le message pour demander des rapports d'activité, activez le gestionnaire de files d'attente pour l'enregistrement des activités et assurez-vous que les applications exécutant des activités sur le message sont capables de générer des rapports d'activité.

Si vous ne souhaitez pas que des rapports d'activité soient générés pour un message lorsqu'il est acheminé via un gestionnaire de files d'attente, *désactivez* le gestionnaire de files d'attente pour l'enregistrement des activités.

## Procédure

1. Demander des rapports d'activité pour un message
  - a) Dans le descripteur de message du message, spécifiez MQRO\_ACTIVITY dans la zone *Rapport* .
  - b) Dans le descripteur de message du message, indiquez le nom d'une file d'attente de réponse dans la zone *ReplyToQ* .

**Avertissement :** Evitez d'activer tous les messages d'un réseau de gestionnaires de files d'attente pour l'enregistrement de l'activité. Les messages activés pour l'enregistrement d'activité peuvent avoir de nombreux rapports d'activité générés en leur nom. Si chaque message d'un réseau de gestionnaires de files d'attente est activé pour l'enregistrement de l'activité, le trafic réseau du gestionnaire de files d'attente peut augmenter jusqu'à un niveau inacceptable.

2. Activez ou désactivez le gestionnaire de files d'attente pour l'enregistrement des activités.

Utilisez la commande MQSC ALTER QMGR, en spécifiant le paramètre ACTIVREC, pour modifier la valeur de l'attribut de gestionnaire de files d'attente. La valeur peut être :

## MSG

Le gestionnaire de files d'attente est activé pour l'enregistrement des activités. Tous les rapports d'activité générés sont distribués à la file d'attente de réponse indiquée dans le descripteur de message du message. Il s'agit de la valeur par défaut.

## QUEUE

Le gestionnaire de files d'attente est activé pour l'enregistrement des activités. Tous les rapports d'activité générés sont distribués à la file d'attente du système local SYSTEM.ADMIN.ACTIVITY.QUEUE. La file d'attente système peut également être utilisée pour transmettre des rapports d'activité à une file d'attente commune.

## DEACTIVE

Le gestionnaire de files d'attente est désactivé pour l'enregistrement d'activité. Aucun rapport d'activité n'est généré dans la portée de ce gestionnaire de files d'attente.

Par exemple, pour activer un gestionnaire de files d'attente pour l'enregistrement d'activité et spécifier que les rapports d'activité générés sont distribués à la file d'attente système locale SYSTEM.ADMIN.ACTIVITY.QUEUE, utilisez la commande MQSC suivante:

```
ALTER QMGR ACTIVREC(QUEUE)
```

**A faire :** Lorsque vous modifiez l'attribut de gestionnaire de files d'attente *ACTIVREC*, un agent MCA en cours d'exécution ne détecte pas la modification tant que le canal n'est pas redémarré.

3. Vérifiez que votre application utilise le même algorithme que celui utilisé par les agents MCA pour déterminer si un rapport d'activité doit être généré pour un message:
  - a) Vérifiez que le message a demandé la génération de rapports d'activité
  - b) Vérifiez que le gestionnaire de files d'attente dans lequel réside le message est activé pour l'enregistrement d'activité
  - c) Placez le rapport d'activité dans la file d'attente déterminée par l'attribut de gestionnaire de files d'attente *ACTIVREC*

## **Configuration d'une file d'attente commune pour les rapports d'activité**

Pour déterminer les emplacements des rapports d'activité associés à un message spécifique lorsque les rapports sont distribués à la file d'attente du système local, il est plus efficace d'utiliser une file d'attente commune sur un noeud unique

## **Avant de commencer**

Définissez le paramètre **ACTIVREC** pour activer le gestionnaire de files d'attente pour l'enregistrement des activités et pour indiquer que les rapports d'activité générés sont distribués à la file d'attente du système local SYSTEM.ADMIN.ACTIVITY.QUEUE.

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

Si un certain nombre de gestionnaires de files d'attente d'un réseau de gestionnaires de files d'attente sont définis pour distribuer des rapports d'activité à la file d'attente du système local, il peut être long de déterminer les emplacements des rapports d'activité associés à un message spécifique. Vous pouvez également utiliser un noeud unique, qui est un gestionnaire de files d'attente qui héberge une file d'attente commune. Tous les gestionnaires de files d'attente d'un réseau de gestionnaires de files d'attente peuvent fournir des rapports d'activité à cette file d'attente commune. L'avantage de l'utilisation d'une file d'attente commune est que les gestionnaires de files d'attente n'ont pas à distribuer de rapports d'activité à la file d'attente de réponse spécifiée dans un message et que, lors de la détermination des emplacements des rapports d'activité associés à un message, vous n'interrogez qu'une seule file d'attente.

Pour configurer une file d'attente commune, procédez comme suit:

## Procédure

1. Sélectionnez ou définissez un gestionnaire de files d'attente comme noeud unique
2. Sur le noeud unique, sélectionnez ou définissez une file d'attente à utiliser comme file d'attente commune
3. Sur tous les gestionnaires de files d'attente où les rapports d'activité doivent être distribués à la file d'attente commune, redéfinissez la file d'attente système locale SYSTEM.ADMIN.ACTIVITY.QUEUE en tant que définition de file d'attente éloignée:
  - a) Indiquez le nom du noeud unique comme nom de gestionnaire de files d'attente éloignées
  - b) Indiquez le nom de la file d'attente commune comme nom de file d'attente éloignée

## Détermination des informations de routage de message

Pour déterminer une route de message, obtenez les informations à partir des rapports d'activité collectés. Déterminez si la file d'attente de réponse contient suffisamment de rapports d'activité pour vous permettre de déterminer les informations requises et d'organiser les rapports d'activité dans l'ordre.

## Pourquoi et quand exécuter cette tâche

L'ordre dans lequel les rapports d'activité sont placés dans la file d'attente de réponses n'est pas nécessairement corrélé à l'ordre dans lequel les activités ont été effectuées. Vous devez classer les rapports d'activité manuellement, sauf s'ils sont générés pour un message de suivi, auquel cas vous pouvez utiliser l'application d'affichage de route IBM MQ pour classer les rapports d'activité.

Déterminez si la file d'attente de réponse contient suffisamment de rapports d'activité pour vous permettre d'obtenir les informations nécessaires:

## Procédure

1. Identifiez tous les rapports d'activité associés dans la file d'attente de réponse en comparant les identificateurs des rapports d'activité et le message d'origine. Veillez à définir l'option de rapport du message d'origine de sorte que les rapports d'activité puissent être corrélés avec le message d'origine.
2. Commandez les rapports d'activité identifiés dans la file d'attente de réponses.  
Vous pouvez utiliser les paramètres suivants à partir du rapport d'activité:

### **OperationType**

Les types d'opérations effectuées peuvent vous permettre de déterminer le rapport d'activité généré directement avant ou après le rapport d'activité en cours.

Par exemple, un rapport d'activité indique qu'un agent MCA a envoyé un message à partir d'une file d'attente de transmission sur un canal. La dernière opération détaillée dans le rapport d'activités comporte un *OperationType* send et des détails indiquant que le message a été envoyé via le canal CH1 au gestionnaire de files d'attente de destination, QM1. Cela signifie que l'activité suivante effectuée sur le message aura eu lieu sur le gestionnaire de files d'attente, QM1, et qu'elle aura commencé avec une opération receive du canal, CH1. A l'aide de ces informations, vous pouvez identifier le prochain rapport d'activité, à condition qu'il existe et qu'il ait été acquis.

### **OperationDate et OperationTime**

Vous pouvez déterminer l'ordre général des activités à partir des dates et des heures des opérations dans chaque rapport d'activité.

**Avertissement :** A moins que les horloges système de chaque gestionnaire de files d'attente du réseau du gestionnaire de files d'attente ne soient synchronisées, l'ordre par date et heure ne garantit pas que les rapports d'activité sont dans l'ordre correct. Vous devez établir la commande manuellement.

L'ordre des rapports d'activité représente la route, ou route partielle, empruntée par le message via le réseau du gestionnaire de files d'attente.

3. Obtenez les informations dont vous avez besoin à partir des informations d'activité dans les rapports d'activité commandés.

Si vous ne disposez pas d'informations suffisantes sur le message, vous pouvez peut-être obtenir d'autres rapports d'activité.

### **Extraction d'autres rapports d'activité**

Pour déterminer une route de message, des informations suffisantes doivent être disponibles à partir des rapports d'activité collectés. Si vous extrayez les rapports d'activité associés à un message de la file d'attente de réponse indiquée, mais que vous ne disposez pas des informations nécessaires, recherchez d'autres rapports d'activité.

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

Pour déterminer les emplacements des autres rapports d'activité, procédez comme suit:

#### **Procédure**

1. Pour tous les gestionnaires de files d'attente du réseau de gestionnaires de files d'attente qui fournissent des rapports d'activité à une file d'attente commune, extrayez les rapports d'activité de la file d'attente commune dont l'ID *CorrelId* correspond à l'ID *MsgId* du message d'origine.
2. Pour tous les gestionnaires de files d'attente du réseau de gestionnaires de files d'attente qui ne fournissent pas de rapports d'activité à une file d'attente commune, extrayez les rapports d'activité comme suit:
  - a) Examinez les rapports d'activité existants pour identifier les gestionnaires de files d'attente par lesquels le message a été acheminé.
  - b) Pour ces gestionnaires de files d'attente, identifiez les gestionnaires de files d'attente activés pour l'enregistrement d'activité.
  - c) Pour ces gestionnaires de files d'attente, identifiez ceux qui n'ont pas renvoyé de rapports d'activité à la file d'attente de réponses spécifiée.
  - d) Pour chacun des gestionnaires de files d'attente que vous identifiez, vérifiez la file d'attente système SYSTEM.ADMIN.ACTIVITY.QUEUE et extrayez tous les rapports d'activité ayant un *CorrelId* qui correspond à l' *MsgId* du message d'origine.
  - e) Si vous ne trouvez aucun rapport d'activité dans la file d'attente système, vérifiez la file d'attente des messages non livrés du gestionnaire de files d'attente, le cas échéant.  
Un rapport d'activités ne peut être distribué dans une file d'attente de rebut que si l'option de rapport, MQRO\_DEAD\_LETTER\_Q, est définie.
3. Organisez tous les rapports d'activités acquis dans l'ordre.  
L'ordre des rapports d'activité représente alors la route, ou route partielle, empruntée par le message.
4. Obtenez les informations dont vous avez besoin à partir des informations d'activité dans les rapports d'activité commandés.  
Dans certains cas, les informations d'activité enregistrées ne peuvent pas atteindre la file d'attente de réponse spécifiée, une file d'attente commune ou une file d'attente système.

### **Circonstances dans lesquelles les informations sur les activités ne sont pas acquises**

Pour déterminer la séquence complète des activités effectuées pour le compte d'un message, les informations relatives à chaque activité doivent être acquises. Si les informations relatives à une activité n'ont pas été enregistrées ou n'ont pas été acquises, vous ne pouvez déterminer qu'une séquence partielle d'activités.

Les informations sur les activités ne sont pas enregistrées dans les cas suivants:

- Le message est traité par un gestionnaire de files d'attente antérieur à IBM WebSphere MQ 6.0.
- Le message est traité par un gestionnaire de files d'attente qui n'est pas activé pour l'enregistrement d'activité.
- L'application qui devait traiter le message n'est pas en cours d'exécution.

Les informations d'activité enregistrées ne peuvent pas atteindre la file d'attente de réponses spécifiée dans les cas suivants:

- Aucun canal n'est défini pour acheminer les rapports d'activité vers la file d'attente de réponse.
- Le canal permettant d'acheminer les rapports d'activité vers la file d'attente de réponse n'est pas en cours d'exécution.
- La définition de file d'attente éloignée utilisée pour acheminer les rapports d'activité vers le gestionnaire de files d'attente où réside la file d'attente de réponse (alias du gestionnaire de files d'attente) n'est pas définie.
- L'utilisateur qui a généré le message d'origine ne dispose pas des droits d'ouverture ou d'insertion sur l'alias du gestionnaire de files d'attente.
- L'utilisateur qui a généré le message d'origine ne dispose pas des droits d'ouverture ou d'insertion sur la file d'attente de réponse.
- La file d'attente de réponse est interdite.

Les informations d'activité enregistrées ne peuvent pas atteindre la file d'attente système ou une file d'attente commune dans les cas suivants:

- Si une file d'attente commune doit être utilisée et qu'aucun canal n'est défini pour acheminer les rapports d'activité vers la file d'attente commune.
- Si une file d'attente commune doit être utilisée et que le canal permettant d'acheminer les rapports d'activité vers la file d'attente commune n'est pas en cours d'exécution.
- Si une file d'attente commune doit être utilisée et que la file d'attente système n'est pas définie correctement.
- L'utilisateur qui a généré le message d'origine ne dispose pas des droits d'ouverture ou d'insertion sur la file d'attente système.
- L'insertion de la file d'attente système est interdite.
- Si une file d'attente commune doit être utilisée et que l'utilisateur qui a généré le message d'origine ne dispose pas des droits d'ouverture ou d'insertion sur la file d'attente commune.
- Si une file d'attente commune doit être utilisée et que l'insertion de la file d'attente commune est interdite.

Dans ces circonstances, si l'option MQRO\_DISCARD\_MSG n'est pas spécifiée pour le rapport d'activité, le rapport d'activité peut être extrait d'une file d'attente de rebut si une file d'attente a été définie sur le gestionnaire de files d'attente où le rapport d'activité a été rejeté. Un rapport d'activité ne comporte cette option de rapport que si le message d'origine, à partir duquel le rapport d'activité a été généré, contient à la fois MQRO\_PASS\_DISCARD\_AND\_EXPIRATION et MQRO\_DISCARD\_MSG dans la zone Rapport du descripteur de message.

## Messagerie de routage de trace

La messagerie d'acheminement de trace est une technique qui utilise des *messages d'acheminement de trace* pour enregistrer les informations d'activité d'un message. La messagerie d'acheminement de trace implique l'envoi d'un message d'acheminement de trace dans un réseau de gestionnaires de files d'attente.

Lorsque le message de trace-route est acheminé via le réseau du gestionnaire de files d'attente, les informations d'activité sont enregistrées. Ces informations d'activité incluent des informations sur les applications qui ont effectué les activités, leur date d'exécution et les opérations qui ont été effectuées dans le cadre des activités. Vous pouvez utiliser les informations enregistrées à l'aide de la messagerie de trace-route pour les raisons suivantes:

### Pour déterminer le dernier emplacement connu d'un message

Si un message n'atteint pas sa destination prévue, vous pouvez utiliser les informations d'activité enregistrées pour un message de routage de trace afin de déterminer le dernier emplacement connu du message. Un message de route de trace est envoyé dans un réseau de gestionnaires de files d'attente avec la même destination cible que le message d'origine, avec l'intention de suivre la même

route. Les informations d'activité peuvent être accumulées dans les données de message du message de suivi ou enregistrées à l'aide de rapports d'activité. Pour augmenter la probabilité que le message de routage de trace suive la même route que le message d'origine, vous pouvez modifier le message de routage de trace pour imiter le message d'origine.

### **Pour déterminer les problèmes de configuration liés à un réseau de gestionnaires de files d'attente**

Les messages de trace-route sont envoyés dans un réseau de gestionnaires de files d'attente et les informations d'activité sont enregistrées. En étudiant les informations d'activité enregistrées pour un message de trace-route, il peut apparaître que le message de trace-route n'a pas suivi la route attendue. Il existe de nombreuses raisons pour lesquelles cela peut se produire, par exemple, un canal peut être inactif, forçant le message à prendre une autre route. Dans ces situations, un administrateur système peut déterminer s'il y a des problèmes dans le réseau du gestionnaire de files d'attente et, le cas échéant, les corriger.

Vous pouvez utiliser l'application d'affichage de route IBM MQ pour configurer, générer et placer des messages de trace dans un réseau de gestionnaires de files d'attente.

**Avvertissement :** Si vous placez un message trace-route dans une liste de distribution, les résultats ne sont pas définis.

### **Concepts associés**

«Référence du message de trace-route», à la page 131

Cette page permet d'obtenir une vue d'ensemble du format des messages de trace. Les données de message de trace-route incluent des paramètres qui décrivent les activités causées par le message de trace-route

### **Mode d'enregistrement des informations d'activité**

Avec la messagerie d'acheminement de trace, vous pouvez enregistrer des informations d'activité dans les données de message du message d'acheminement de trace ou utiliser des rapports d'activité. Vous pouvez également utiliser les deux techniques.

### **Accumulation des informations d'activité dans les données de message du message de suivi**

Lorsqu'un message de trace-route est acheminé via un réseau de gestionnaires de files d'attente, les informations relatives aux activités effectuées pour le compte du message de trace-route peuvent être cumulées dans les données de message du message de trace-route. Les informations d'activité sont stockées dans les groupes PCF *Activity*. Pour chaque activité effectuée pour le compte du message de suivi, un groupe PCF *Activity* est écrit à la fin du bloc PCF dans les données de message du message de suivi.

Des informations d'activité supplémentaires sont enregistrées dans la messagerie trace-route, dans un groupe PCF appelé groupe PCF *TraceRoute*. Les informations d'activité supplémentaires sont stockées dans ce groupe PCF et peuvent être utilisées pour déterminer la séquence des activités enregistrées. Cette technique est contrôlée par le paramètre *Accumuler* du groupe PCF *TraceRoute*.

### **Enregistrement d'informations d'activité à l'aide de rapports d'activité**

Lorsqu'un message de trace-route est acheminé via un réseau de gestionnaires de files d'attente, un rapport d'activité peut être généré pour chaque activité effectuée pour le compte du message de trace-route. Les informations d'activité sont stockées dans le groupe PCF *Activity*. Pour chaque activité effectuée pour le compte d'un message de suivi, un rapport d'activité est généré contenant un groupe PCF *Activity*. L'enregistrement d'activité pour les messages de trace-route fonctionne de la même manière que pour tout autre message.

Les rapports d'activité générés pour les messages de trace-route contiennent des informations d'activité supplémentaires par rapport à celles générées pour tout autre message. Les informations supplémentaires sont renvoyées dans un groupe PCF *TraceRoute*. Les informations contenues dans le groupe PCF *TraceRoute* ne sont exactes qu'à partir du moment où le rapport d'activité a été généré. Vous pouvez utiliser les informations supplémentaires pour déterminer la séquence des activités effectuées pour le compte du message de suivi.



## **Acquisition d'informations sur les activités enregistrées**

Lorsqu'un message de trace-route a atteint sa destination prévue ou est supprimé, la méthode que vous utilisez pour obtenir les informations d'activité dépend de la façon dont ces informations ont été enregistrées.

### **Avant de commencer**

Si vous ne connaissez pas les informations sur l'activité, voir [«Mode d'enregistrement des informations d'activité»](#), à la page 72.

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

Utilisez les méthodes suivantes pour acquérir les informations d'activité une fois que le message de trace-route a atteint sa destination prévue ou qu'il a été supprimé:

#### **Procédure**

- Extrayez le message de trace-route.

Le paramètre *Distribution*, dans le groupe PCF *TraceRoute*, détermine si un message de trace est placé dans la file d'attente cible à l'arrivée ou s'il est supprimé. Si le message de trace-route est distribué à la file d'attente cible, vous pouvez extraire le message de trace-route de cette file d'attente. Ensuite, vous pouvez utiliser l'application d'affichage de route IBM MQ pour afficher les informations d'activité.

Pour demander que les informations d'activité soient accumulées dans les données de message d'un message de routage de trace, définissez le paramètre *Accumulation* dans le groupe PCF *TraceRoute* sur MQRROUTE\_ACCUMULATE\_IN\_MSG.

- Utilisez un message de réponse de trace-route.

Lorsqu'un message de trace-route atteint sa destination prévue ou que le message de trace-route ne peut plus être acheminé dans un réseau de gestionnaires de files d'attente, un message de réponse de trace-route peut être généré. Un message de réponse de trace-route contient un doublon de toutes les informations d'activité du message de trace-route et est distribué à une file d'attente de réponse spécifiée ou à la file d'attente système SYSTEM.ADMIN.TRACE.ROUTE.QUEUE. Vous pouvez utiliser l'application d'affichage de route IBM MQ pour afficher les informations d'activité.

Pour demander un message de réponse de trace-route, définissez le paramètre *Accumuler* dans le groupe PCF *TraceRoute* sur MQRROUTE\_ACCUMULATE\_AND\_REPLY.

- Utilisez les rapports d'activité.

Si des rapports d'activité sont générés pour un message trace-route, vous devez localiser les rapports d'activité avant de pouvoir obtenir les informations d'activité. Ensuite, pour déterminer la séquence des activités, vous devez classer les rapports d'activité.

### **Contrôle de la messagerie de routage de trace**

Activez la messagerie de routage de trace au niveau du gestionnaire de files d'attente, de sorte que les applications dans la portée de ce gestionnaire de files d'attente puissent écrire des informations d'activité dans un message de routage de trace. Pour activer l'intégralité d'un réseau de gestionnaires de files d'attente, activez individuellement chaque gestionnaire de files d'attente du réseau pour la messagerie de suivi. Si vous activez davantage de gestionnaires de files d'attente, des rapports d'activité supplémentaires sont générés.

### **Avant de commencer**

Si vous utilisez des rapports d'activité pour enregistrer des informations d'activité pour un message de trace-route, voir [«Contrôle de l'enregistrement des activités»](#), à la page 67.

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

Pour enregistrer les informations d'activité d'un message de routage de trace lors de son acheminement via un gestionnaire de files d'attente, procédez comme suit:

## Procédure

- Définissez la façon dont les informations d'activité doivent être enregistrées pour le message de trace-route.  
Voir «Génération et configuration d'un message de trace-route», à la page 76
- Si vous souhaitez accumuler des informations d'activité dans le message de trace-route, vérifiez que le gestionnaire de files d'attente est activé pour la messagerie de trace-route
- Si vous souhaitez accumuler des informations d'activité dans le message de trace-route, assurez-vous que les applications exécutant des activités sur le message de trace-route sont capables d'écrire des informations d'activité dans les données de message du message de trace-route

## Concepts associés

«Génération et configuration d'un message de trace-route», à la page 76

Un message de trace-route comprend un descripteur de message spécifique et des parties de données de message. Pour générer un message de trace-route, créez le message manuellement ou utilisez l'application d'affichage de route IBM MQ .

## Tâches associées

«Contrôle de l'enregistrement des activités», à la page 67

Activez l'enregistrement des activités au niveau du gestionnaire de files d'attente. Pour activer l'intégralité d'un réseau de gestionnaires de files d'attente, activez individuellement chaque gestionnaire de files d'attente du réseau pour l'enregistrement de l'activité. Si vous activez davantage de gestionnaires de files d'attente, des rapports d'activité supplémentaires sont générés.

### *Activation des gestionnaires de files d'attente pour la messagerie de trace-route*

Pour contrôler si les gestionnaires de files d'attente sont activés ou désactivés pour la messagerie d'acheminement de trace, utilisez l'attribut de gestionnaire de files d'attente ROUTEREC.

Utilisez la commande MQSC ALTER QMGRen spécifiant le paramètre ROUTEREC pour modifier la valeur de l'attribut de gestionnaire de files d'attente. La valeur peut être l'une des valeurs suivantes :

## MSG

Le gestionnaire de files d'attente est activé pour la messagerie de suivi. Les applications dans la portée du gestionnaire de files d'attente peuvent écrire des informations d'activité dans le message de trace.

Si le paramètre *Accumulation* du groupe PCF *TraceRoute* est défini sur MQROUTE\_ACCUMULATE\_AND\_REPLYet que l'activité suivante à effectuer sur le message de trace-route est la suivante:

- est une suppression
- est une insertion dans une file d'attente locale (file d'attente cible ou file d'attente de rebut)
- Le nombre total d'activités effectuées sur le message trace-route dépasse la valeur du paramètre *MaxActivities*, dans le groupe PCF *TraceRoute* .

un message de réponse de trace-route est généré et distribué à la file d'attente de réponse spécifiée dans le descripteur de message du message de trace-route.

## QUEUE

Le gestionnaire de files d'attente est activé pour la messagerie de suivi. Les applications dans la portée du gestionnaire de files d'attente peuvent écrire des informations d'activité dans le message de trace.

Si le paramètre *Accumulation* du groupe PCF *TraceRoute* est défini sur MQROUTE\_ACCUMULATE\_AND\_REPLYet que l'activité suivante à effectuer sur le message de trace-route est la suivante:

- est une suppression
- est une insertion dans une file d'attente locale (file d'attente cible ou file d'attente de rebut)
- Le nombre total d'activités effectuées sur le message trace-route dépasse la valeur du paramètre *MaxActivities*, dans le groupe PCF *TraceRoute* .

un message de réponse de trace-route est généré et distribué à la file d'attente du système local SYSTEM.ADMIN.TRACE.ROUTE.QUEUE.

## DESACTIVE

Le gestionnaire de files d'attente est désactivé pour la messagerie de trace-route. Les informations d'activité ne sont pas cumulées dans le message de trace, mais le groupe PCF *TraceRoute* peut être mis à jour dans la portée de ce gestionnaire de files d'attente.

Par exemple, pour désactiver un gestionnaire de files d'attente pour la messagerie d'acheminement de trace, utilisez la commande MQSC suivante:

```
ALTER QMGR ROUTEREC(DISABLED)
```

**A faire :** Lorsque vous modifiez l'attribut de gestionnaire de files d'attente *ROUTEREC*, un agent MCA en cours d'exécution ne détecte pas la modification tant que le canal n'est pas redémarré.

### Activation des applications pour la messagerie de trace-route

Pour activer la messagerie de suivi pour une application utilisateur, basez votre algorithme sur l'algorithme utilisé par les agents MCA (Message Channel Agent)

## Avant de commencer

Si vous n'êtes pas familiarisé avec le format d'un message trace-route, voir [«Référence du message de trace-route»](#), à la page 131.

## Pourquoi et quand exécuter cette tâche

Les agents MCA (Message Channel Agent) sont activés pour la messagerie de suivi. Pour activer une application utilisateur pour la messagerie d'acheminement de trace, procédez comme suit à partir de l'algorithme utilisé par les agents MCA:

## Procédure

1. Déterminez si le message en cours de traitement est un message de suivi.  
Si le message n'est pas conforme au format d'un message de trace-route, il n'est pas traité comme un message de trace-route.
2. Déterminez si les informations d'activité doivent être enregistrées.  
Si le niveau de détail de l'activité effectuée n'est pas inférieur au niveau de détail spécifié par le paramètre *Détails*, les informations sur l'activité sont enregistrées dans des circonstances spécifiques. Ces informations sont enregistrées uniquement si le message de trace route demande une accumulation et que le gestionnaire de files d'attente est activé pour la messagerie de trace route, ou si le message de trace route demande un rapport d'activité et que le gestionnaire de files d'attente est activé pour l'enregistrement d'activité.
  - Si des informations d'activité doivent être enregistrées, incrémentez le paramètre *RecordedActivities*.
  - Si les informations d'activité ne doivent pas être enregistrées, incrémentez le paramètre *UnrecordedActivities*.
3. Déterminez si le nombre total d'activités effectuées sur le message de trace-route dépasse la valeur du paramètre *MaxActivities*.  
Le nombre total d'activités correspond à la somme de *RecordedActivities*, *UnrecordedActivities* et *DiscontinuityCount*.  
Si le nombre total d'activités dépasse *MaxActivities*, rejetez le message avec des commentaires en retour MQFB\_MAX\_ACTIVITÉS.
4. Si la valeur de *Accumulation* est définie sur MQROUTE\_ACCUMULATE\_IN\_MSG ou MQROUTE\_ACCUMULATE\_AND\_REPLY et que le gestionnaire de files d'attente est activé pour la

messagerie d'acheminement de trace, écrivez un groupe PCF d'activité à la fin du bloc PCF dans les données de message d'un message d'acheminement de trace.

5. Distribuez le message de trace-route dans une file d'attente locale.

- Si le paramètre *Distribution* est défini sur MQROUTE\_DELIVER\_NO, rejetez le message de suivi MQFB\_NOT\_DELIVERED.
- Si le paramètre *Distribution* est spécifié en tant que MQROUTE\_DELIVER\_YES, le message de trace est envoyé à la file d'attente locale.

6. Générez un message de réponse de trace-route si toutes les conditions suivantes sont remplies:

- Le message de trace-route a été distribué dans une file d'attente locale ou a été rejeté
- La valeur du paramètre, *Accumuler*, est MQROUTE\_ACCUMULATE\_AND\_REPLY
- Le gestionnaire de files d'attente est activé pour la messagerie de trace-route

Le message de réponse de suivi est inséré dans la file d'attente déterminée par l'attribut de gestionnaire de files d'attente ROUTEREC.

7. Si le message de trace-route a demandé un rapport d'activité et que le gestionnaire de files d'attente est activé pour l'enregistrement d'activité, générez un rapport d'activité.

Le rapport d'activités est placé dans la file d'attente déterminée par l'attribut de gestionnaire de files d'attente ACTIVREC.

### **Génération et configuration d'un message de trace-route**

Un message de trace-route comprend un descripteur de message spécifique et des parties de données de message. Pour générer un message de trace-route, créez le message manuellement ou utilisez l'application d'affichage de route IBM MQ .

Un message de trace-route se compose des éléments suivants:

#### **Descripteur de message**

Une structure MQMD, avec la zone *Format* définie sur MQFMT\_ADMIN ou MQFMT\_EMBEDDED\_PCF.

#### **Données de message**

Une des combinaisons suivantes:

- Un en-tête PCF (MQCFH) et des données de message de trace, si *Format* est défini sur MQFMT\_ADMIN
- Un en-tête PCF imbriqué (MQEPH), des données de message de trace et des données de message supplémentaires spécifiées par l'utilisateur, si *Format* est défini sur MQFMT\_EMBEDDED\_PCF

Les données de message de trace sont constituées du groupe PCF *TraceRoute* et d'un ou de plusieurs groupes PCF *Activity* .

### **Génération manuelle**

Lors de la génération manuelle d'un message trace-route, un groupe PCF *Activity* n'est pas requis. *Activité* Les groupes PCF sont écrits dans les données de message du message de suivi lorsqu'un agent MCA ou une application écrite par l'utilisateur effectue une activité en son nom.

### **L'application d'affichage de route IBM MQ**

Utilisez l'application de routage d'affichage IBM MQ , **dspmqrte**, pour configurer, générer et placer un message de routage de trace dans un réseau de gestionnaires de files d'attente. Définissez le paramètre *Format* dans le descripteur de message sur MQFMT\_ADMIN. Vous ne pouvez pas ajouter de données utilisateur au message de trace-route généré par l'application d'affichage de route IBM MQ .

**Restriction :** dspmqrte ne peut pas être émis sur les gestionnaires de files d'attente IBM MQ for z/OS . Si vous souhaitez que le premier gestionnaire de files d'attente via lequel le message de trace est acheminé soit un gestionnaire de files d'attente de ce type, connectez-vous au gestionnaire de files d'attente en tant que client à l'aide du paramètre facultatif - c.

### Imitant le message d'origine

Lors de l'utilisation d'un message de trace-route pour déterminer la route empruntée par un autre message via un réseau de gestionnaires de files d'attente, plus un message de trace-route imite le message d'origine, plus la probabilité que le message de trace-route suive la même route que le message d'origine est élevée.

Les caractéristiques de message suivantes peuvent affecter l'emplacement de réacheminement d'un message dans un réseau de gestionnaires de files d'attente:

#### **Priorité**

La priorité peut être spécifiée dans le descripteur de message du message.

#### **Persistance**

La persistance peut être spécifiée dans le descripteur de message du message.

#### **Expiration**

L'expiration peut être indiquée dans le descripteur de message du message.

#### **Options de rapport**

Les options de rapport peuvent être spécifiées dans le descripteur de message du message.

#### **Taille de message**

Pour imiter la taille d'un message, des données supplémentaires peuvent être écrites dans les données du message. A cet effet, des données de message supplémentaires peuvent être insignifiantes.

**Conseil :** L'application d'affichage de route IBM MQ ne peut pas spécifier de taille de message.

#### **Données de message**

Certains réseaux de gestionnaires de files d'attente utilisent le routage basé sur le contenu pour déterminer où les messages sont transférés. Dans ces cas, les données de message du message de trace doivent être écrites pour imiter les données de message du message d'origine.

**Conseil :** L'application d'affichage de route IBM MQ ne peut pas spécifier de données de message.

### Groupe PCF TraceRoute

Les attributs du groupe PCF *TraceRoute* contrôlent le comportement d'un message de suivi. Le groupe PCF *TraceRoute* se trouve dans les données de message de chaque message de suivi.

Le tableau suivant répertorie les paramètres du groupe *TraceRoute* reconnus par un agent MCA. D'autres paramètres peuvent être ajoutés si des applications écrites par l'utilisateur sont écrites pour les reconnaître, comme décrit dans «Informations supplémentaires sur l'activité», à la page 82.

Paramètre	Tapez
TraceRoute	MQCFGR
Détails	MQCFIN
RecordedActivities	MQCFIN
UnrecordedActivities	MQCFIN
DiscontinuityCount	MQCFIN
MaxActivities	MQCFIN
Cumuler	MQCFIN
Faire suivre	MQCFIN
Délivrer	MQCFIN

Les descriptions de chaque paramètre du groupe PCF *TraceRoute* sont les suivantes:

#### **Détails**

Indique le niveau de détail des informations d'activité à enregistrer. La valeur peut être l'une des valeurs suivantes :

#### **MQROUTE\_DETAIL\_LOW**

Seules les activités effectuées par l'application utilisateur sont enregistrées.

## **MQRROUTE\_DETAIL\_MEDIUM**

Les activités spécifiées dans MQRROUTE\_DETAIL\_LOW doivent être enregistrées. A celles-ci s'ajoutent les activités effectuées par les agents MCA.

## **MQRROUTE\_DETAIL\_ELEVE**

Les activités spécifiées dans MQRROUTE\_DETAIL\_LOW et MQRROUTE\_DETAIL\_MEDIUM doivent être enregistrées. Les MCM n'enregistrent aucune autre information sur l'activité à ce niveau de détail. Cette option est disponible uniquement pour les applications utilisateur qui doivent enregistrer des informations d'activité supplémentaires. Par exemple, si une application utilisateur détermine la route empruntée par un message en prenant en compte certaines caractéristiques de message, les informations relatives à la logique de routage peuvent être incluses avec ce niveau de détail.

### **RecordedActivities**

Indique le nombre d'activités enregistrées effectuées pour le compte du message de suivi. Une activité est considérée comme étant enregistrée si des informations la concernant ont été écrites dans le message de suivi ou si un rapport d'activité a été généré. Pour chaque activité enregistrée, *RecordedActivities* est incrémenté de un.

### **UnrecordedActivities**

Indique le nombre d'activités non enregistrées effectuées pour le compte du message de suivi. Une activité est considérée comme n'étant pas enregistrée si une application qui est activée pour la messagerie de routage de trace ne s'accumule pas et n'écrit pas les informations d'activité associées dans un rapport d'activité.

Une activité effectuée pour le compte d'un message trace-route n'est pas enregistrée dans les cas suivants:

- Le niveau de détail de l'activité effectuée est inférieur au niveau de détail spécifié par le paramètre *Détail*.
- Le message de trace-route demande un rapport d'activité, mais pas d'accumulation, et le gestionnaire de files d'attente n'est pas activé pour l'enregistrement d'activité.
- Le message de trace-route demande une accumulation, mais pas un rapport d'activité, et le gestionnaire de files d'attente n'est pas activé pour la messagerie de trace-route.
- Le message de trace-route demande à la fois une accumulation et un rapport d'activité, et le gestionnaire de files d'attente n'est pas activé pour l'enregistrement d'activité et la messagerie de trace-route.
- Le message de trace-route ne demande ni accumulation, ni rapport d'activité.

Pour chaque activité non enregistrée, le paramètre *UnrecordedActivities* est incrémenté de un.

### **DiscontinuityCount**

Indique le nombre de fois où le message de trace-route a été acheminé via un gestionnaire de files d'attente avec des applications qui n'étaient pas activées pour la messagerie de trace-route. Cette valeur est incrémentée par le gestionnaire de files d'attente. Si cette valeur est supérieure à 0, seule une route de message partielle peut être déterminée.

### **MaxActivities**

Indique le nombre maximal d'activités pouvant être effectuées pour le compte du message de suivi.

Le nombre total d'activités correspond à la somme de *RecordedActivities*, *UnrecordedActivities* et *DiscontinuityCount*. Le nombre total d'activités ne doit pas dépasser la valeur de *MaxActivities*.

La valeur de *MaxActivities* peut être:

#### **Entier positif**

Nombre maximal d'activités.

Si le nombre maximal d'activités est dépassé, le message de trace-route est rejeté avec un retour d'informations MQFB\_MAX\_ACTIVITÉS. Cela peut empêcher le message de trace-route d'être réacheminé indéfiniment s'il est intercepté dans une boucle sans fin.

## **MQRROUTE\_UNLIMITED\_ACTIVITÉS**

Un nombre illimité d'activités peut être effectué pour le compte du message de suivi.

### **Cumuler**

Indique la méthode utilisée pour accumuler les informations d'activité. La valeur peut être l'une des valeurs suivantes :

#### **MQRROUTE\_ACCUMULATE\_EN\_MSG**

Si le gestionnaire de files d'attente est activé pour la messagerie d'acheminement de trace, les informations d'activité sont accumulées dans les données de message du message d'acheminement de trace.

Si cette valeur est spécifiée, les données du message de trace sont les suivantes:

- Groupe PCF *TraceRoute* .
- Zéro ou plusieurs groupes PCF *Activity* .

#### **MQRROUTE\_ACCUMULATE\_ET\_REPONSE**

Si le gestionnaire de files d'attente est activé pour la messagerie de trace-route, les informations d'activité sont accumulées dans les données de message du message de trace-route et un message de réponse de trace-route est généré si l'une des situations suivantes se produit:

- Le message de suivi est supprimé par un gestionnaire de files d'attente IBM MQ .
- Le message de suivi est inséré dans une file d'attente locale (file d'attente cible ou file d'attente de rebut) par un gestionnaire de files d'attente IBM MQ .
- Le nombre d'activités effectuées sur le message de trace-route dépasse la valeur de *MaxActivities*.

Si cette valeur est spécifiée, les données du message de trace sont les suivantes:

- Groupe PCF *TraceRoute* .
- Zéro ou plusieurs groupes PCF *Activity* .

#### **MQRROUTE\_ACCUMULATE\_NONE**

Les informations d'activité ne sont pas accumulées dans les données de message du message de suivi.

Si cette valeur est spécifiée, les données du message de trace sont les suivantes:

- Groupe PCF *TraceRoute* .

### **Faire suivre**

Indique à quel endroit un message de trace-route peut être acheminé. La valeur peut être :

#### **MQRROUTE\_WARD\_IF\_SUPPORTED**

Le message de trace-route est réacheminé uniquement vers les gestionnaires de files d'attente qui respecteront la valeur du paramètre *Distribution* du groupe *TraceRoute* .

#### **MQRROUTE\_ACHEMINEMENT\_TOUT**

Le message de trace-route est réacheminé vers n'importe quel gestionnaire de files d'attente, que la valeur du paramètre *Distribution* soit ou non respectée.

Les gestionnaires de files d'attente utilisent l'algorithme suivant pour déterminer si un message de trace doit être acheminé vers un gestionnaire de files d'attente éloignées:

1. Déterminez si le gestionnaire de files d'attente éloignées est capable de prendre en charge la messagerie de suivi.
  - Si le gestionnaire de files d'attente éloignées est capable de prendre en charge la messagerie de suivi, l'algorithme passe à l'étape «4», à la page 80.
  - Si le gestionnaire de files d'attente éloignées n'est pas capable de prendre en charge la messagerie de suivi, l'algorithme passe à l'étape «2», à la page 79
2. Déterminez si le paramètre *Distribution* du groupe *TraceRoute* contient des options de distribution non reconnues dans le masque de bit MQRROUTE\_DELIVER\_REJ\_UNSUP\_MASK.

- Si des options de distribution non reconnues sont trouvées, le message de trace est rejeté avec un retour d'informations MQFB\_UNSUPPORTED\_DISTRIBUTION.
  - Si aucune option de distribution non reconnue n'est trouvée, l'algorithme passe à l'étape «3», à la page 80.
3. Déterminez la valeur du paramètre *Distribuer* à partir du groupe PCF *TraceRoute* dans le message trace-route.
    - Si l'option *Distribuer* est définie sur MQROUTE\_DELIVER\_YES, le message de trace est réacheminé vers le gestionnaire de files d'attente éloignées.
    - Si *Distribution* est spécifié en tant que MQROUTE\_DELIVER\_NO, l'algorithme passe à l'étape «4», à la page 80.
  4. Déterminez si le paramètre *Forward* du groupe *TraceRoute* contient des options de réacheminement non reconnues dans le masque de bits MQROUTE\_FORWARD\_ing\_rej\_unsup\_mask.
    - Si des options de réacheminement non reconnues sont trouvées, le message trace-route est rejeté avec le retour d'informations MQFB\_UNSUPPORTED\_FORWARDING.
    - Si aucune option de transfert non reconnue n'est trouvée, l'algorithme passe à l'étape «5», à la page 80.
  5. Déterminez la valeur du paramètre *Transférer* à partir du groupe PCF *TraceRoute* dans le message de trace-route.
    - Si *Forward* est spécifié comme MQROUTE\_FORWARD\_if\_supported, le message de trace-route est rejeté avec les commentaires en retour MQFB\_NOT\_FORWARD.
    - Si *Forward* est spécifié sous la forme MQROUTE\_FORWARD\_all, le message de trace-route peut être réacheminé vers le gestionnaire de files d'attente éloignées.

### **Délivrer**

Indique l'action à entreprendre si le message de trace-route atteint la destination prévue. Les applications écrites par l'utilisateur doivent vérifier cet attribut avant de placer un message de trace-route dans sa file d'attente cible. La valeur peut être l'une des valeurs suivantes :

#### **MQROUTE\_DELIVER\_YES**

A l'arrivée, le message de suivi est placé dans la file d'attente cible. Toute application effectuant une opération d'extraction sur la file d'attente cible peut extraire le message de trace.

#### **MQROUTE\_NO**

A l'arrivée, le message de trace-route n'est pas distribué dans la file d'attente cible. Le message est traité en fonction de ses options de rapport.

### **Configuration d'une file d'attente commune pour les messages de réponse d'acheminement de trace**

Pour déterminer les emplacements des messages de réponse de routage liés à un message spécifique lorsque les rapports sont distribués à la file d'attente du système local, il est plus efficace d'utiliser une file d'attente commune sur un noeud unique

### **Avant de commencer**

Définissez le paramètre **ROUTEREC** pour activer le gestionnaire de files d'attente pour la messagerie d'acheminement de trace et pour indiquer que tous les messages de réponse d'acheminement de trace générés sont distribués à la file d'attente système locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE.

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

Si un certain nombre de gestionnaires de files d'attente d'un réseau de gestionnaires de files d'attente sont définis pour distribuer des messages de réponse d'acheminement de trace à la file d'attente du système local, il peut être long de déterminer les emplacements des messages de réponse d'acheminement de trace associés à un message spécifique. Vous pouvez également utiliser un noeud unique, qui est un gestionnaire de files d'attente qui héberge une file d'attente commune. Tous les



gestionnaires de files d'attente d'un réseau de gestionnaires de files d'attente peuvent distribuer des messages de réponse d'acheminement de trace à cette file d'attente commune. L'avantage de l'utilisation d'une file d'attente commune est que les gestionnaires de files d'attente n'ont pas besoin de distribuer des messages de réponse d'acheminement de trace à la file d'attente de réponse spécifiée dans un message et que, lors de la détermination des emplacements des messages de réponse d'acheminement de trace associés à un message, vous n'interrogez qu'une seule file d'attente.

Pour configurer une file d'attente commune, procédez comme suit:

## Procédure

1. Sélectionnez ou définissez un gestionnaire de files d'attente comme noeud unique
2. Sur le noeud unique, sélectionnez ou définissez une file d'attente à utiliser comme file d'attente commune
3. Sur tous les gestionnaires de files d'attente qui réacheminent les messages de réponse de trace vers la file d'attente commune, redéfinissez la file d'attente système locale SYSTEM.ADMIN.TRACE.ROUTE.QUEUE en tant que définition de file d'attente éloignée
  - a) Indiquez le nom du noeud unique comme nom de gestionnaire de files d'attente éloignées
  - b) Indiquez le nom de la file d'attente commune comme nom de file d'attente éloignée

## Acquisition et utilisation des informations enregistrées

Utilisez l'une des techniques suivantes pour acquérir des informations d'activité enregistrées pour un message de suivi

Notez que les circonstances dans lesquelles les informations d'activité ne sont pas acquises s'appliquent également aux messages de réponse de routage.

Les informations d'activité ne sont pas enregistrées lorsqu'un message de trace-route est traité par un gestionnaire de files d'attente qui est désactivé pour l'enregistrement d'activité et la messagerie de trace-route.

### *Acquisition d'informations à partir de messages de réponse de trace-route*

Pour obtenir des informations sur l'activité, recherchez le message de réponse de trace-route. Ensuite, vous extrayez le message et analysez les informations d'activité.

## Pourquoi et quand exécuter cette tâche

Vous ne pouvez acquérir des informations d'activité à partir d'un message de réponse de trace-route que si vous connaissez l'emplacement du message de réponse de trace-route. Localisez le message et traitez les informations d'activité comme suit:

## Procédure

1. Vérifiez la file d'attente de réponse spécifiée dans le descripteur de message du message de suivi. Si le message de réponse de trace-route ne se trouve pas dans la file d'attente de réponse, vérifiez les emplacements suivants:
  - File d'attente du système local, SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, sur le gestionnaire de files d'attente cible du message de trace-route
  - La file d'attente commune, si vous avez configuré une file d'attente commune pour les messages de réponse de routage de trace
  - File d'attente du système local, SYSTEM.ADMIN.TRACE.ROUTE.QUEUE, sur tout autre gestionnaire de files d'attente du réseau de gestionnaires de files d'attente, ce qui peut se produire si le message de trace-route a été inséré dans une file d'attente de rebut ou si le nombre maximal d'activités a été dépassé
2. Extraire le message de réponse de trace-route
3. Utilisez l'application d'affichage de route IBM MQ pour afficher les informations d'activité enregistrées
4. Etudiez les informations sur l'activité et obtenez les informations dont vous avez besoin

### *Acquisition d'informations à partir de messages de trace-route*

Pour acquérir des informations d'activité, vous devez localiser le message de trace-route, qui doit avoir les paramètres appropriés dans le groupe PCF *TraceRoute* . Ensuite, vous extrayez le message et analysez les informations d'activité.

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

Vous pouvez acquérir des informations d'activité à partir d'un message de trace-route uniquement si vous connaissez l'emplacement du message de trace-route et qu'il comporte le paramètre *Accumuler* dans le groupe PCF *TraceRoute* spécifié sous la forme MQRROUTE\_ACCUMULATE\_IN\_MSG ou MQRROUTE\_ACCUMULATE\_AND\_REPLY.

Pour que le message de trace-route soit distribué à la file d'attente cible, le paramètre *Distribution* du groupe PCF *TraceRoute* doit être défini sur MQRROUTE\_DELIVER\_YES.

### **Procédure**

1. Vérifiez la file d'attente cible. Si le message de trace-route ne se trouve pas dans la file d'attente cible, vous pouvez essayer de localiser le message de trace-route à l'aide d'un message de trace-route activé pour l'enregistrement d'activité. Avec les rapports d'activité générés, essayez de déterminer le dernier emplacement connu du message de trace.
2. Extraire le message de trace-route
3. Utilisez l'application d'affichage de route IBM MQ pour afficher les informations d'activité enregistrées
4. Etudiez les informations sur l'activité et obtenez les informations dont vous avez besoin

### *Acquisition d'informations à partir de rapports d'activité*

Pour obtenir des informations sur les activités, recherchez le rapport d'activités, qui doit comporter l'option de rapport indiquée dans le descripteur de message. Ensuite, vous extrayez le rapport d'activité et vous analysez les informations d'activité.

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

Vous ne pouvez acquérir des informations d'activité à partir d'un rapport d'activité que si vous connaissez l'emplacement du rapport d'activité et que l'option de rapport MQRORO\_ACTIVITY a été spécifiée dans le descripteur de message du message de trace-route.

### **Procédure**

1. Recherchez et ordonnez les rapports d'activité générés pour un message de suivi.  
Une fois que vous avez localisé les rapports d'activité, vous pouvez les commander manuellement ou utiliser l'application IBM MQ display route pour commander et afficher automatiquement les informations d'activité.
2. Etudiez les informations sur l'activité et obtenez les informations dont vous avez besoin

### **Informations supplémentaires sur l'activité**

Lorsqu'un message de trace est acheminé via un réseau de gestionnaires de files d'attente, les applications utilisateur peuvent enregistrer des informations supplémentaires en incluant un ou plusieurs paramètres PCF supplémentaires lors de l'écriture du groupe *Activité* dans les données de message du message de trace ou du rapport d'activité.

Des informations d'activité supplémentaires peuvent aider les administrateurs système à identifier la route empruntée par un message de trace-route ou la raison pour laquelle cette route a été empruntée.

Si vous utilisez l'application d'affichage de route IBM MQ pour afficher les informations enregistrées pour un message de trace, tous les paramètres PCF supplémentaires ne peuvent être affichés qu'avec un identificateur numérique, sauf si l'identificateur de chaque paramètre est reconnu par l'application d'affichage de route IBM MQ . Pour reconnaître un identificateur de paramètre, des informations supplémentaires doivent être enregistrées à l'aide des paramètres PCF suivants. Incluez ces paramètres PCF à un emplacement approprié dans le groupe PCF *Activité* .

## GroupName

Tableau 13. Nom du groupe	
Description	Paramètres groupés spécifiant les informations supplémentaires.
ID	MQGACF_VALEUR_NAMING.
Type de données	MQCFGR
Paramètres du groupe	<i>ParameterName</i> <i>ParameterValue</i>

## ParameterName

Tableau 14. Nom du paramètre :	
Description	Contient le nom à afficher par l'application IBM MQ display route, qui place la valeur de <i>ParameterValue</i> en contexte.
ID	MQCA_VALEUR_NAME.
Type de données	MQCFST
Inclus dans le groupe PCF:	<i>GroupName.</i>
Valeur :	Nom à afficher.

## ParameterValue

Tableau 15. Valeur du paramètre	
Description	Contient la valeur à afficher par l'application d'affichage de route IBM MQ .
Identificateur :	Identificateur de structure PCF pour les informations supplémentaires.
Type de données :	Type de données de structure PCF pour les informations supplémentaires.
Inclus dans le groupe PCF:	<i>GroupName.</i>
Valeur :	Valeur à afficher.

## Exemples d'enregistrement d'informations d'activité supplémentaires

Les exemples suivants illustrent comment une application utilisateur peut enregistrer des informations supplémentaires lors de l'exécution d'une activité pour le compte d'un message de suivi. Dans les deux exemples, l'application IBM MQ display route est utilisée pour générer un message trace-route et afficher les informations d'activité qui lui sont renvoyées.

### Enregistrement d'informations supplémentaires sur l'activité: Exemple 1

Les informations d'activité supplémentaires sont enregistrées par une application utilisateur dans un format où l'identificateur de paramètre *n'est pas* reconnu par l'application d'affichage de route IBM MQ .

1. L'application d'affichage de route IBM MQ permet de générer et d'insérer un message de trace dans un réseau de gestionnaires de files d'attente. Les options nécessaires sont définies pour demander les éléments suivants:
  - Les informations d'activité sont accumulées dans les données de message du message de suivi.
  - A l'arrivée dans la file d'attente cible, le message de suivi est supprimé et un message de réponse de suivi est généré et distribué dans une file d'attente de réponse spécifiée.
  - A la réception du message de réponse de l'acheminement de trace, l'application IBM MQ affiche les informations d'activité cumulées.

Le message de trace-route est inséré dans le réseau du gestionnaire de files d'attente.

2. Lorsque le message de trace-route est acheminé via le réseau du gestionnaire de files d'attente, une application utilisateur, qui est activée pour la messagerie de trace-route, effectue une activité de détail faible pour le compte du message. Outre l'écriture des informations d'activité standard dans le message de suivi, l'application utilisateur écrit le paramètre PCF suivant à la fin du groupe d'activités:

#### **ColorValue**

##### **ID**

65536

##### **Type de données**

MQCFST

##### **Valeur**

'Rouge'

Ce paramètre PCF supplémentaire fournit des informations supplémentaires sur l'activité qui a été effectuée, mais il est écrit dans un format dans lequel l'identificateur de paramètre *n'est pas* reconnu par l'application d'affichage de route IBM MQ .

3. Les messages de trace-route atteignent la file d'attente cible et un message de réponse de trace-route est renvoyé à l'application d'affichage de route IBM MQ . Les informations supplémentaires sur l'activité s'affichent comme suit:

```
65536: 'Red'
```

L'application d'affichage de route IBM MQ ne reconnaît pas l'identificateur de paramètre du paramètre PCF et l'affiche sous forme de valeur numérique. Le contexte des informations supplémentaires n'est pas clair.

Pour un exemple de cas où l'application d'affichage de route IBM MQ reconnaît l'identificateur de paramètre du paramètre PCF, voir [«Enregistrement d'informations supplémentaires sur l'activité: Exemple 2»](#), à la page 84.

#### *Enregistrement d'informations supplémentaires sur l'activité: Exemple 2*

Les informations d'activité supplémentaires sont enregistrées par une application utilisateur dans un format où l'identificateur de paramètre est reconnu par l'application d'affichage de route IBM MQ .

1. L'application d'affichage de route IBM MQ permet de générer et d'insérer un message de trace dans un réseau de gestionnaires de files d'attente de la même manière que dans [«Enregistrement d'informations supplémentaires sur l'activité: Exemple 1»](#), à la page 83.
2. Lorsque le message de trace-route est acheminé via le réseau du gestionnaire de files d'attente, une application utilisateur, qui est activée pour la messagerie de trace-route, effectue une activité de détail faible pour le compte du message. En plus d'écrire les informations d'activité standard dans le message de suivi, l'application utilisateur écrit les paramètres PCF suivants à la fin du groupe d'activités:

#### **ColorInfo**

<i>Tableau 16. Informations de couleur</i>	
<b>Description</b>	<b>Paramètres groupés spécifiant des informations sur une couleur.</b>
Identificateur :	MQGACF_VALEUR_NAMING.
Type de données :	MQCFGR.
Paramètres du groupe:	<i>ColorName</i> <i>ColorValue</i>

## ColorName

Tableau 17. Nom de la couleur	
<b>Description</b>	<b>Contient le nom à afficher par l'application IBM MQ display route qui place la valeur de ColorValue en contexte.</b>
Identificateur :	MQCA_VALEUR_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	ColorInfo.
Valeur :	'Couleur'

## ColorValue

Tableau 18. Valeur de couleur	
<b>Description</b>	<b>Contient la valeur à afficher par l'application d'affichage de route IBM MQ .</b>
Identificateur :	65536.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	ColorInfo.
Valeur :	'Rouge'


Ces paramètres PCF supplémentaires fournissent des informations supplémentaires sur l'activité qui a été effectuée. Ces paramètres PCF sont écrits dans un format dans lequel l'identificateur de paramètre est reconnu par l'application d'affichage de route IBM MQ .

3. Les messages de trace-route atteignent la file d'attente cible et un message de réponse de trace-route est renvoyé à l'application d'affichage de route IBM MQ . Les informations supplémentaires sur l'activité s'affichent comme suit:

```
Color: 'Red'
```

L'application d'affichage de route IBM MQ reconnaît que l'identificateur de paramètre de la structure PCF contenant la valeur des informations d'activité supplémentaires a un nom correspondant. Le nom correspondant est affiché à la place de la valeur numérique.

## IBM MQ afficher l'application de route

Utilisez l'application d'affichage de route IBM MQ ( **dspmqrte** ) pour utiliser des messages de trace-route et des informations d'activité liées à un message de trace-route, à l'aide d'une interface de ligne de commande.  L'application de route d'affichage IBM MQ n'est pas fournie avec IBM MQ for z/OS, mais vous pouvez l'exécuter à partir d'une installation répartie et la connecter en tant que client à un gestionnaire de files d'attente IBM MQ for z/OS en spécifiant le paramètre **-c** lors de l'émission de la commande **dspmqrte** .

Vous pouvez utiliser l'application de route d'affichage IBM MQ [dspmqrte](#) dans les buts suivants:

- Pour configurer, générer et placer un message de trace dans un réseau de gestionnaires de files d'attente.

En insérant un message de trace-route dans un réseau de gestionnaires de files d'attente, vous pouvez collecter des informations d'activité et les utiliser pour déterminer la route empruntée par le message de trace-route. Vous pouvez spécifier les caractéristiques des messages trace-route comme suit:

- Destination du message de suivi.
- Comment le message de trace-route imite un autre message.
- Comment le message de trace-route doit être traité lorsqu'il est acheminé via un réseau de gestionnaires de files d'attente.
- Indique si la messagerie d'enregistrement d'activité ou de suivi est utilisée pour enregistrer les informations d'activité.
- Permet de commander et d'afficher des informations d'activité liées à un message de trace-route.

Si l'application d'affichage de route IBM MQ a inséré un message de trace dans un réseau de gestionnaires de files d'attente, une fois que les informations d'activité associées ont été renvoyées, les informations peuvent être commandées et affichées immédiatement. L'application IBM MQ display route peut également être utilisée pour commander et afficher les informations d'activité liées à un message trace-route précédemment généré.

### Référence associée

[dspmqrte](#)

### Paramètres des messages de trace-route

Cette page permet d'obtenir une vue d'ensemble des paramètres fournis par l'application IBM MQ display route, **dspmqrte**, pour déterminer les caractéristiques d'un message trace-route, y compris la manière dont il est traité lorsqu'il est acheminé via un réseau de gestionnaires de files d'attente.

### Référence associée

[dspmqrte](#)

#### Connexion de gestionnaire de files d'attente

Cette page permet de spécifier le gestionnaire de files d'attente auquel l'application de route d'affichage IBM MQ se connecte.

#### -c

Indique que l'application de route d'affichage IBM MQ se connecte en tant qu'application client.

Si vous ne spécifiez pas ce paramètre, l'application de route d'affichage IBM MQ ne se connecte pas en tant qu'application client.

#### -m *NomGestF*

Nom du gestionnaire de files d'attente auquel l'application de route d'affichage IBM MQ se connecte. Vous pouvez indiquer 48 caractères au maximum.

Si vous ne spécifiez pas ce paramètre, le gestionnaire de files d'attente par défaut est utilisé.

#### Destination cible

Cette page permet de spécifier la destination cible d'un message de suivi

#### -q *NomFileCible*

Si l'application IBM MQ display route est utilisée pour envoyer un message de trace-route à un réseau de gestionnaires de files d'attente, *TargetQName* indique le nom de la file d'attente cible.

#### -ts *TargetTopicChaîne*

Indique la chaîne de rubrique.

#### -qm *TargetQMgr*

Qualifie la destination cible ; la résolution de nom de gestionnaire de files d'attente normale s'applique alors. La destination cible est spécifiée avec *-q TargetQName* ou *-ts TargetTopicString* .

Si vous ne spécifiez pas ce paramètre, le gestionnaire de files d'attente auquel l'application de route d'affichage IBM MQ est connectée est utilisé comme gestionnaire de files d'attente cible.

#### -o

Indique que la destination cible n'est pas liée à une destination spécifique. Ce paramètre est normalement utilisé lorsque le message de suivi doit être placé dans un cluster. La destination cible est ouverte avec l'option MQOO\_BIND\_NOT\_FIXED.

Si vous ne spécifiez pas ce paramètre, la destination cible est liée à une destination spécifique.

#### *Rubrique de publication*

Pour les applications de publication / abonnement, utilisez cette page pour spécifier la chaîne de rubrique d'un message de trace pour l'application d'affichage de route IBM MQ à publier

#### **-ts TopicName**

Indique une chaîne de rubrique dans laquelle l'application de route d'affichage IBM MQ doit publier un message de trace-route et place cette application en mode rubrique. Dans ce mode, l'application trace tous les messages qui résultent de la demande de publication.

Vous pouvez également utiliser l'application d'affichage de route IBM MQ pour afficher les résultats d'un rapport d'activité généré pour les messages de publication.

#### *Imitateur de message*

Cette page permet de configurer un message de suivi pour imiter un message, par exemple lorsque le message d'origine n'a pas atteint sa destination prévue.

L'une des utilisations de la messagerie de trace-route consiste à déterminer le dernier emplacement connu d'un message qui n'a pas atteint sa destination prévue. L'application d'affichage de route IBM MQ fournit des paramètres qui peuvent aider à configurer un message de trace-route pour imiter le message d'origine. Lorsque vous imitez un message, vous pouvez utiliser les paramètres suivants:

#### **-l Persistence**

Indique la persistance du message de suivi généré. Les valeurs possibles pour *Persistence* sont les suivantes:

##### **oui**

Le message de suivi généré est persistant. (MQPER\_PERSISTENT).

##### **non**

Le message de trace-route généré n'est **pas** persistant. (MQPER\_NOT\_PERSISTENT).

##### **q**

Le message de trace-route généré hérite de la valeur de persistance de la destination spécifiée par *-q TargetQName* ou *-ts TargetTopicString*. (MQPER\_PERSISTENCE\_AS\_Q\_DEF).

Un message de réponse de trace-route ou tout message de rapport renvoyé partagera la même valeur de persistance que le message de trace-route d'origine.

Si *Persistence* est défini sur **yes**, vous devez spécifier le paramètre *-rq ReplyToQ*. La file d'attente de réponse ne doit pas être résolue en file d'attente dynamique temporaire.

Si vous ne spécifiez pas ce paramètre, le message de trace généré est **non** persistant.

#### **-p Priorité**

Indique la priorité du message de suivi. La valeur *Priorité* est supérieure ou égale à 0, ou MQPRI\_PRIORITY\_AS\_Q\_DEF. MQPRI\_PRIORITY\_AS\_Q\_DEF indique que la valeur de priorité est extraite de la destination spécifiée par *-q TargetQName* ou *-ts TargetTopicString*.

Si vous ne spécifiez pas ce paramètre, la valeur de priorité est extraite de la destination spécifiée par *-q TargetQName* ou *-ts TargetTopicString*.

#### **-xs Expiration**

Indique le délai d'expiration du message de suivi, en secondes.

Si vous ne spécifiez pas ce paramètre, le délai d'expiration est de 60 secondes.

#### **-ro none |ReportOption**

##### **none**

Indique qu'aucune option de rapport n'est définie.

##### **ReportOption**

Indique les options de rapport pour le message de suivi. Plusieurs options de rapport peuvent être spécifiées à l'aide d'une virgule comme séparateur. Les valeurs possibles pour *ReportOption* sont les suivantes:

**activité**

L'option de rapport MQRO\_ACTIVITY est définie.

**coa**

L'option de rapport MQRO\_COA\_WITH\_FULL\_DATA est définie.

**cod**

L'option de rapport MQRO\_COD\_WITH\_FULL\_DATA est définie.

**exception**

L'option de rapport MQRO\_EXCEPTION\_WITH\_FULL\_DATA est définie.

**expiration**

L'option de rapport MQRO\_EXPIRATION\_WITH\_FULL\_DATA est définie.

**annuler**

L'option de rapport MQRO\_DISCARD\_MSG est définie.

Si ni *-ro ReportOption* ni *-ro none* ne sont spécifiés, les options de rapport MQRO\_ACTIVITY et MQRO\_DISCARD\_MSG sont spécifiées.

L'application d'affichage de route IBM MQ ne vous permet pas d'ajouter des données utilisateur au message de trace-route. Si vous avez besoin que des données utilisateur soient ajoutées au message de trace-route, vous devez générer le message de trace-route manuellement.

*Informations sur les activités enregistrées*

Utilisez cette page pour spécifier la méthode utilisée pour renvoyer les informations d'activité enregistrées, que vous pouvez ensuite utiliser pour déterminer la route empruntée par un message de trace-route

Les informations d'activité enregistrées peuvent être renvoyées comme suit:

- Dans les rapports d'activité
- Dans un message de réponse de suivi
- Dans le message de trace-route lui-même (ayant été placé dans la file d'attente cible)

Lors de l'utilisation de **dspmqrte**, la méthode utilisée pour renvoyer les informations d'activité enregistrées est déterminée à l'aide des paramètres suivants:

**-ro activité**

Indique que les informations d'activité sont renvoyées à l'aide des rapports d'activité. Par défaut, l'enregistrement d'activité est activé.

**-ac -ar**

Indique que les informations d'activité sont accumulées dans le message de suivi et qu'un message de réponse de suivi doit être généré.

**-ac**

Indique que les informations d'activité doivent être accumulées dans le message de suivi (trace-route).

Si vous ne spécifiez pas ce paramètre, ces informations ne sont pas accumulées dans ce message.

**-ar**

Demande qu'un message de réponse de suivi contenant toutes les informations d'activité cumulées soit généré dans les cas suivants:

- Le message de suivi est supprimé par un gestionnaire de files d'attente IBM MQ .
- Le message de suivi est inséré dans une file d'attente locale (file d'attente cible ou file d'attente de rebut) par un gestionnaire de files d'attente IBM MQ .
- Le nombre d'activités effectuées sur le message de suivi dépasse la valeur indiquée dans *-s Activities*.

**-ac -d oui**

Indique que les informations d'activité sont accumulées dans le message de routage de trace et qu'à leur arrivée, le message de routage de trace est placé dans la file d'attente cible.



**-ac**

Indique que les informations d'activité doivent être accumulées dans le message de suivi (trace-route).

Si vous ne spécifiez pas ce paramètre, ces informations ne sont pas accumulées dans ce message.

**-d oui**

A l'arrivée, le message de trace-route est inséré dans la file d'attente cible, même si le gestionnaire de files d'attente ne prend pas en charge la messagerie de trace-route.

Si vous ne spécifiez pas ce paramètre, le message de suivi n'est pas placé dans la file d'attente cible.

Le message de trace-route peut ensuite être extrait de la file d'attente cible et les informations d'activité enregistrées peuvent être acquises.

Vous pouvez combiner ces méthodes selon vos besoins.

En outre, le niveau de détail des informations d'activité enregistrées peut être spécifié à l'aide du paramètre suivant:

**-t *Détail***

Indique les activités qui sont enregistrées. Les valeurs possibles pour *Détails* sont les suivantes:

**bas**

Seules les activités effectuées par une application définie par l'utilisateur sont enregistrées.

**support**

Les activités spécifiées en bas sont enregistrées. En outre, les activités de publication et les activités effectuées par les MCM sont enregistrées.

**haut**

Les activités spécifiées dans les niveaux bas et moyen sont enregistrées. Les agents MCA n'exposent pas d'autres informations d'activités à ce niveau de détail. Cette option est disponible pour les applications définies par l'utilisateur qui doivent exposer d'autres informations d'activité uniquement. Par exemple, si une application définie par l'utilisateur détermine la route qu'emprunte un message en prenant en compte certaines caractéristiques de message, la logique de routage peut être incluse avec ce niveau de détail.

Si vous ne spécifiez pas ce paramètre, les activités de niveau moyen sont enregistrées.

Par défaut, l'application de route d'affichage IBM MQ utilise une file d'attente dynamique temporaire pour stocker les messages renvoyés. Lorsque l'application d'affichage de route IBM MQ se termine, la file d'attente dynamique temporaire est fermée et tous les messages sont purgés. Si les messages renvoyés sont requis au-delà de l'exécution en cours de l'application IBM MQ display route, une file d'attente permanente doit être spécifiée à l'aide des paramètres suivants:

**-rq *FileRéponse***

Indique le nom de la file d'attente de réponse à laquelle sont envoyés tous les messages de suivi. Si le message de suivi est persistant, ou si le paramètre *-n* est spécifié, une file d'attente de réponse autre qu'une file d'attente dynamique temporaire doit être spécifiée.

Si vous ne spécifiez pas ce paramètre, une file d'attente de réponse dynamique est créée à l'aide de la file d'attente modèle par défaut du système, SYSTEM.DEFAULT.MODEL.QUEUE.

**-rqm *GestFileRéponse***

Indique le nom du gestionnaire de files d'attente dans lequel réside la file d'attente de réponse. Vous pouvez indiquer 48 caractères au maximum.

Si vous ne spécifiez pas ce paramètre, le gestionnaire de files d'attente auquel l'application de route d'affichage IBM MQ est connectée est utilisé comme gestionnaire de files d'attente de réponse.

***Mode de traitement du message de trace-route***

Cette page permet de contrôler la façon dont un message de trace est traité lorsqu'il est acheminé via un réseau de gestionnaires de files d'attente.

Les paramètres suivants peuvent restreindre l'emplacement où le message de trace peut être acheminé dans le réseau du gestionnaire de files d'attente:

#### **-d Distribuer**

Indique si le message de suivi doit être distribué à la file d'attente cible à son arrivée. Les valeurs possibles pour *Distribuer* sont les suivantes:

##### **oui**

A l'arrivée, le message de trace-route est inséré dans la file d'attente cible, même si le gestionnaire de files d'attente ne prend pas en charge la messagerie de trace-route.

##### **non**

A son arrivée, le message de suivi n'est pas placé dans la file d'attente cible.

Si vous ne spécifiez pas ce paramètre, le message de suivi n'est pas placé dans la file d'attente cible.

#### **-f Transmettre**

Indique le type de gestionnaire de files d'attente vers lequel le message de suivi peut être transmis. Pour plus de détails sur l'algorithme utilisé par les gestionnaires de files d'attente pour déterminer si un message doit être réacheminé vers un gestionnaire de files d'attente éloignées, voir «Groupe PCF TraceRoute», à la page 77. Les valeurs possibles pour *Transférer* sont les suivantes:

##### **tout**

Le message de suivi est transmis à n'importe quel gestionnaire de files d'attente.

**Avertissement :** S'il est réacheminé vers un gestionnaire de files d'attente antérieur à IBM WebSphere MQ 6.0, le message de trace-route ne sera pas reconnu et pourra être acheminé vers une file d'attente locale malgré la valeur du paramètre *-d Deliver* .

##### **pris en charge**

Le message de trace-route est réacheminé uniquement vers un gestionnaire de files d'attente qui respecte le paramètre *Distribution* du groupe PCF *TraceRoute*

Si vous ne spécifiez pas ce paramètre, le message de trace-route sera réacheminé uniquement vers un gestionnaire de files d'attente qui prendra en charge le paramètre *Distribution* .

Les paramètres suivants peuvent empêcher un message de trace-route de rester indéfiniment dans un réseau de gestionnaire de files d'attente:

#### **-s Activités**

Indique le nombre maximal d'activités enregistrées qui peuvent être effectuées au nom du message de suivi avant qu'il soit supprimé. Ainsi, le message de trace-route ne peut pas être réacheminé indéfiniment s'il est intercepté dans une boucle sans fin. La valeur *Activités* est supérieure ou égale à 1, ou MQROUTE\_UNLIMITED\_ACTIVITIES. MQROUTE\_UNLIMITED\_ACTIVITIES indique qu'un nombre illimité d'activités peuvent être effectuées au nom du message de suivi.

Si vous ne spécifiez pas ce paramètre, un nombre illimité d'activités peuvent être effectuées au nom du message de suivi.

#### **-xs Expiration**

Indique le délai d'expiration du message de suivi, en secondes.

Si vous ne spécifiez pas ce paramètre, le délai d'expiration est de 60 secondes.

#### **-xp TransmettreExpiration**

Indique si le délai d'expiration du message de trace-route est transmis à un message de réponse de trace-route. Les valeurs possibles pour *PassExpiry* sont les suivantes:

##### **oui**

L'option de rapport MQRO\_PASS\_DISCARD\_AND\_EXPIRY est indiquée dans le descripteur du message de suivi.

Si un message de réponse de trace-route ou des rapports d'activité sont générés pour le message de trace-route, l'option de rapport MQRO\_DISCARD (si elle est spécifiée) et le délai d'expiration restant sont transmis.

Il s'agit de la valeur par défaut.

**non**

L'option de rapport MQRO\_PASS\_DISCARD\_AND\_EXPIRY n'est pas spécifiée.

Si un message de réponse de trace-route est généré pour le message de trace-route, l'option de suppression et l'heure d'expiration du message de trace-route ne sont **pas** transmises.

Si vous ne spécifiez pas ce paramètre, MQRO\_PASS\_DISCARD\_AND\_EXPIRATION n'est pas spécifié.

**-ro supprimer**

Indique l'option de rapport MQRO\_DISCARD\_MSG. Cela peut empêcher le message de trace-route de rester indéfiniment dans le réseau du gestionnaire de files d'attente.

**Affichage des informations d'activité**

L'application d'affichage de route IBM MQ peut afficher des informations d'activité pour un message de routage de trace qu'elle vient d'insérer dans un réseau de gestionnaires de files d'attente, ou elle peut afficher des informations d'activité pour un message de routage de trace précédemment généré. Il peut également afficher des informations supplémentaires enregistrées par les applications écrites par l'utilisateur.

Pour indiquer si les informations d'activité renvoyées pour un message de trace-route sont affichées, spécifiez le paramètre suivant:

**-n**

Indique que les informations d'activité pour le message de suivi ne doivent pas s'afficher.

Si ce paramètre est accompagné d'une demande de message de réponse de routage de trace ( *-ar* ) ou d'une des options de génération de rapport à partir de ( *-ro ReportOption* ), une file d'attente de réponse spécifique (non modèle) doit être spécifiée à l'aide de *-rq ReplyToQ* . Par défaut, seuls les messages de rapport d'activité sont demandés.

Une fois le message de trace-route inséré dans la file d'attente cible spécifiée, une chaîne hexadécimale de 48 caractères contenant l'identificateur du message de trace-route s'affiche. L'identificateur de message peut être utilisé par l'application IBM MQ display route pour afficher ultérieurement les informations d'activité du message trace-route, à l'aide du paramètre *-i CorrelId* .

Si vous ne spécifiez pas ce paramètre, les informations d'activité renvoyées pour le message de suivi s'affichent sous la forme spécifiée par le paramètre *-v* .

Lors de l'affichage des informations d'activité d'un message de suivi qui vient d'être inséré dans un réseau de gestionnaires de files d'attente, le paramètre suivant peut être spécifié:

**-w TempsAttente**

Indique la durée, en secondes, pendant laquelle l'application IBM MQ display route attend que les rapports d'activité ou un message de réponse de trace soient renvoyés à la file d'attente de réponse spécifiée.

Si vous ne spécifiez pas ce paramètre, le temps d'attente est défini comme le temps d'expiration du message de suivi, plus 60 secondes.

Lors de l'affichage des informations d'activité précédemment accumulées, les paramètres suivants doivent être définis:

**-q NomFileCible**

Si l'application d'affichage de route IBM MQ est utilisée pour afficher les informations d'activité précédemment collectées, *TargetQName* indique le nom de la file d'attente dans laquelle les informations d'activité sont stockées.

**-i IDcorrélation**

Ce paramètre est utilisé lorsque l'application d'affichage de route IBM MQ est utilisée pour afficher uniquement les informations d'activité précédemment cumulées. Il existe un grand nombre de rapports d'activité et de messages de réponse de suivi (trace-route) dans la file d'attente indiquée par *-q NomFileCible* . *CorrelId* est utilisé pour identifier les rapports d'activité, ou un message de réponse de trace-route, associé à un message de trace-route. Indiquez l'identificateur du message de trace d'origine dans *CorrelId* .

Le format de *CorrelId* est une chaîne hexadécimale de 48 caractères.

Les paramètres suivants peuvent être utilisés lors de l'affichage des informations d'activité cumulées précédemment ou lors de l'affichage des informations d'activité en cours pour un message de suivi:

**-b**

Indique que l'application d'affichage de route IBM MQ ne parcourt que les rapports d'activité ou un message de réponse de trace lié à un message. Cela permet d'afficher à nouveau les informations sur l'activité ultérieurement.

Si vous ne spécifiez pas ce paramètre, l'application d'affichage de route IBM MQ obtient de façon destructive des rapports d'activité ou un message de réponse de routage de trace associé à un message.

**-v summary | all | none | structure *DisplayOption***

**récapitulatif**

Les files d'attente par le biais desquelles est acheminé le message de suivi sont affichées.

**tout**

Toutes les informations disponibles seront affichées.

**Aucun**

Aucune information n'est affichée.

**outline *OptionAffichage***

Indique les options d'affichage pour le message de suivi. Plusieurs options d'affichage peuvent être spécifiées à l'aide d'une virgule comme séparateur.

Si aucune valeur n'est fournie, les éléments suivants s'affichent:

- Nom de l'application
- Type de chaque opération
- Tous les paramètres spécifiques à l'opération

Les valeurs possibles pour *DisplayOption* sont les suivantes:

**activité**

Tous les paramètres de groupe non PCF des groupes PCF *Activité* sont affichés.

**Identificateurs**

Les valeurs associées aux identificateurs de paramètre MQBACF\_MSG\_ID ou MQBACF\_CORREL\_ID sont affichées. Cette valeur remplace *msgdelta*.

**Message**

Tous les paramètres de groupe non PCF des groupes PCF *Message* sont affichés. Lorsque cette valeur est indiquée, *msgdelta* ne peut pas être spécifié.

**msgdelta**

Tous les paramètres de groupe non PCF des groupes PCF *Message* qui ont été modifiés depuis la dernière opération sont affichés. Lorsque cette valeur est indiquée, *message* ne peut pas être spécifié.

**opération**

Tous les paramètres de groupe non PCF des groupes PCF *Opération* sont affichés.

**routage**

Tous les paramètres de groupe non PCF dans les groupes PCF *TraceRoute* sont affichés.

Si vous ne spécifiez pas ce paramètre, un récapitulatif de la route des messages s'affiche.

## Affichage d'informations supplémentaires

Lorsqu'un message de routage de trace est acheminé via un réseau de gestionnaires de files d'attente, les applications écrites par l'utilisateur peuvent enregistrer des informations supplémentaires en écrivant un ou plusieurs paramètres PCF supplémentaires dans les données de message du message de routage de trace ou dans les données de message d'un rapport d'activité. Pour que l'application d'affichage de route IBM MQ affiche des informations supplémentaires dans un format lisible, elle doit être enregistrée dans un format spécifique, comme décrit dans [«Informations supplémentaires sur l'activité», à la page 82.](#)

## Exemples d'application de route d'affichage IBM MQ

Les exemples suivants montrent comment utiliser l'application d'affichage de route IBM MQ. Dans chaque exemple, deux gestionnaires de files d'attente (QM1 et QM2) sont interconnectés par deux canaux (QM2.TO.QM1 et QM1.TO.QM2).

### Exemple 1-Demande de rapports d'activité

Affichage des informations d'activité à partir d'un message de suivi transmis à la file d'attente cible

Dans cet exemple, l'application de route d'affichage IBM MQ se connecte au gestionnaire de files d'attente, QM1, et est utilisée pour générer et distribuer un message de route de trace à la file d'attente cible, TARGET.Q, sur le gestionnaire de files d'attente éloignées, QM2. L'option de rapport nécessaire est spécifiée de sorte que les rapports d'activité soient demandés lors du routage du message de réponse de trace. A l'arrivée dans la file d'attente cible, le message de trace est supprimé. Les informations d'activité renvoyées à l'application d'affichage de route IBM MQ à l'aide des rapports d'activité sont placées dans l'ordre et affichées.

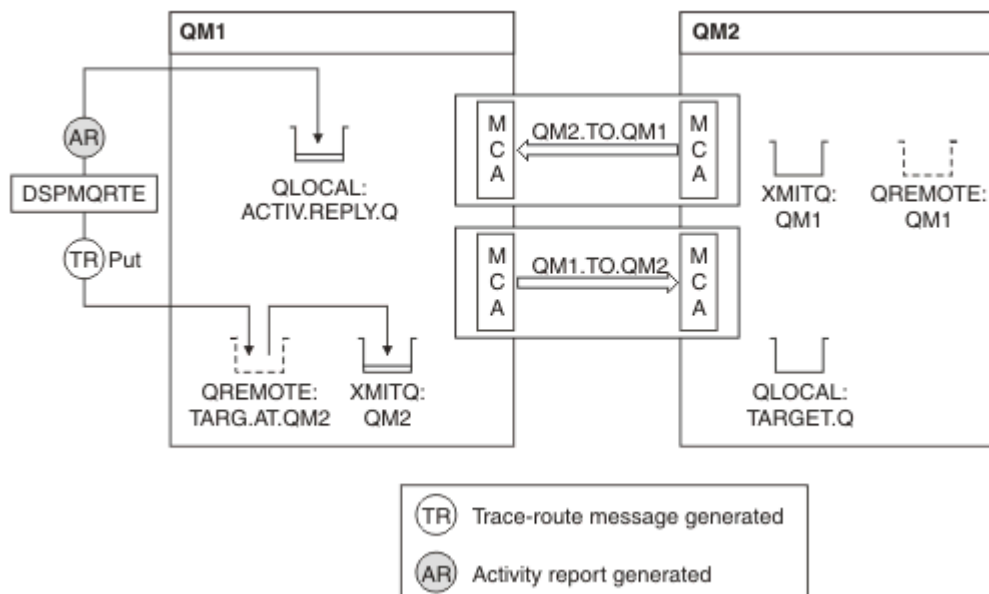


Figure 9. Demande de rapports d'activité, diagramme 1

- L'attribut ACTIVREC de chaque gestionnaire de files d'attente (QM1 et QM2) est défini sur MSG.
- La commande suivante est émise:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq ACTIV.REPLY.Q
```

QM1 est le nom du gestionnaire de files d'attente auquel l'application de route d'affichage IBM MQ se connecte, TARG.AT.QM2 est le nom de la file d'attente cible et ACTIV.REPLY.Q est le nom de la file d'attente à laquelle il est demandé que toutes les réponses au message de suivi soient envoyées.

Les valeurs par défaut sont utilisées pour toutes les options qui ne sont pas spécifiées, mais notez en particulier l'option -f (le message de trace-route est transmis uniquement à un gestionnaire de files d'attente qui respecte le paramètre de distribution du groupe PCF TraceRoute), l'option -d (à l'arrivée, le message de trace-route n'est pas placé dans la file d'attente cible), l'option -ro (les options de rapport MQRO\_ACTIVITY et MQRO\_DISCARD\_MSG sont spécifiées) et l'option -t (l'activité de niveau de détail moyen est enregistrée).

- DSPMQRTE génère le message trace-route et le place dans la file d'attente éloignée TARG.AT.QM2.
- DSPMQRTE examine ensuite la valeur de l'attribut ACTIVREC du gestionnaire de files d'attente QM1. La valeur est MSG. Par conséquent, DSPMQRTE génère un rapport d'activité et le place dans la file d'attente de réponses ACTIV.REPLY.Q.

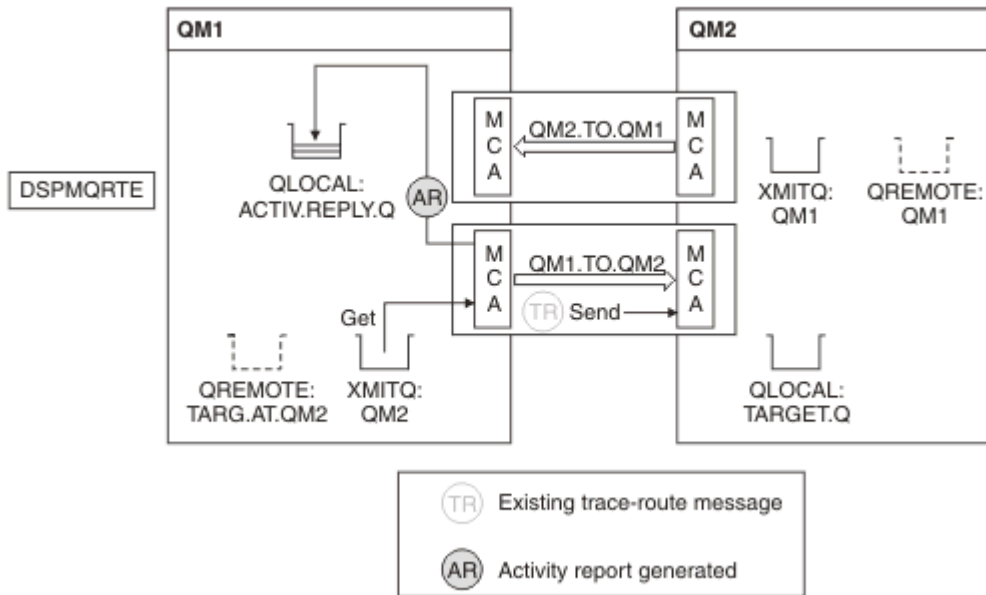


Figure 10. Demande de rapports d'activité, diagramme 2

- L'agent MCA (Message Channel Agent) émetteur extrait le message de trace de la file d'attente de transmission. Le message est un message d'acheminement de trace. Par conséquent, l'agent MCA commence à enregistrer les informations d'activité.
- L'attribut ACTIVREC du gestionnaire de files d'attente (QM1) est MSG et l'option MQRO\_ACTIVITY est spécifiée dans la zone Rapport du descripteur de message. Par conséquent, l'agent MCA générera ultérieurement un rapport d'activité. La valeur du paramètre RecordedActivities du groupe PCF TraceRoute est incrémentée de 1.
- L'agent MCA vérifie que la valeur MaxActivities du groupe PCF TraceRoute n'a pas été dépassée.
- Avant de transmettre le message à QM2, l'agent MCA suit l'algorithme décrit dans la rubrique [Transfert](#) (étapes «1», à la page 79, «4», à la page 80 et «5», à la page 80). et l'agent MCA choisit d'envoyer le message.
- L'agent MCA génère ensuite un rapport d'activité et le place dans la file d'attente de réponses (ACTIV.REPLY.Q).

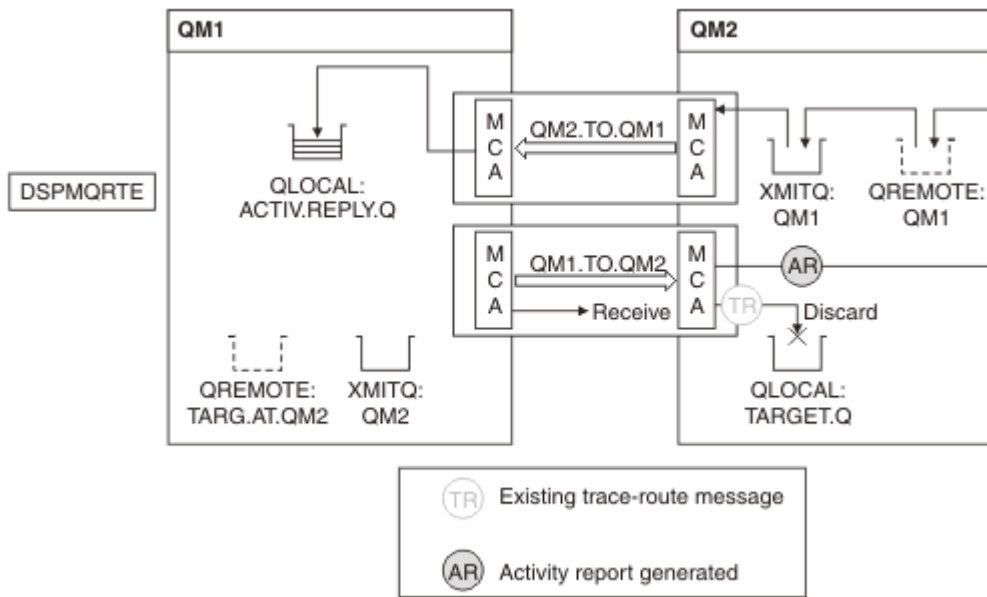


Figure 11. Demande de rapports d'activité, diagramme 3

- L'agent MCA récepteur reçoit le message de suivi du canal. Le message est un message de suivi ; par conséquent, l'agent MCA commence à enregistrer les informations relatives à l'activité.
- Si le gestionnaire de files d'attente d'où provient le message de trace est IBM WebSphere MQ 5.3.1 ou une version antérieure, l'agent MCA incrémente de 1 le paramètre DiscontinuityCount du PCF TraceRoute . Ce n'est pas le cas ici.
- L'attribut ACTIVREC du gestionnaire de files d'attente (QM2) est MSG et l'option MQRO\_ACTIVITY est spécifiée. Par conséquent, l'agent MCA génère un rapport d'activité. La valeur du paramètre RecordedActivities est incrémentée de 1.
- La file d'attente cible est une file d'attente locale. Par conséquent, le message est supprimé avec le retour d'informations MQFB\_NOT\_DELIVERED, conformément à la valeur du paramètre de distribution dans le groupe PCF TraceRoute .
- L'agent MCA génère ensuite le rapport d'activité final et le place dans la file d'attente de réponses. La file d'attente de transmission associée au gestionnaire de files d'attente QM1 est résolue et le rapport d'activité est renvoyé au gestionnaire de files d'attente QM1 (ACTIV.REPLY.Q).

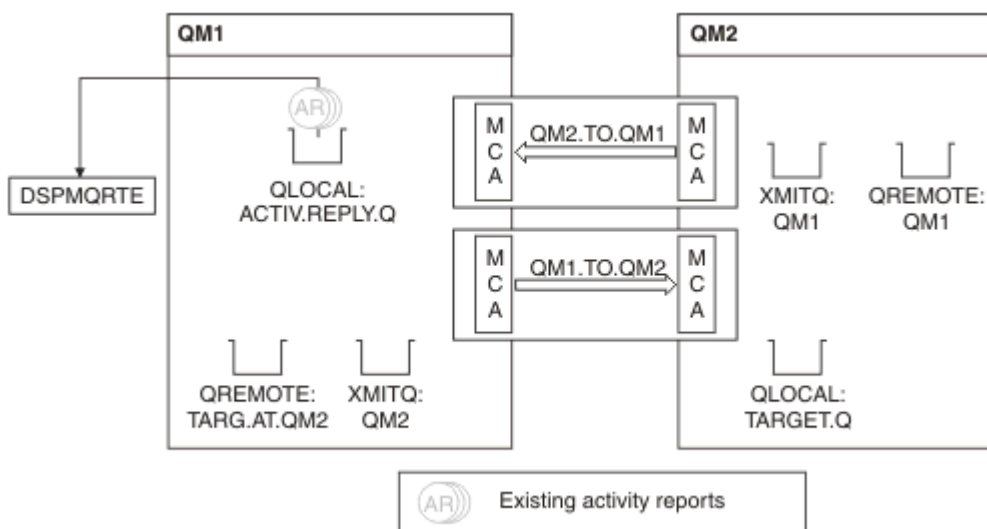


Figure 12. Demande de rapports d'activité, diagramme 4

- Dans le même temps, DSPMQRTE a continuellement exécuté des MQGET sur la file d'attente de réponses (ACTIV.REPLY.Q), en attente de rapports d'activité. Il attend jusqu'à 120 secondes (60 secondes de plus que le délai d'expiration du message de trace) car -w n'a pas été spécifié lors du démarrage de DSPMQRTE.
- La commande DSPMQRTE permet d'obtenir les 3 rapports d'activité de la file d'attente de réponses.
- Les rapports d'activité sont classés à l'aide des paramètres RecordedActivities, UnrecordedActivitieset DiscontinuityCount du groupe PCF TraceRoute pour chacune des activités. La seule valeur différente de zéro dans cet exemple est RecordedActivities. Il s'agit donc du seul paramètre réellement utilisé.
- Le programme s'arrête dès que l'opération de suppression est affichée. Même si l'opération finale a été une opération de suppression, elle est traitée comme si une opération d'insertion avait eu lieu car le retour d'informations est MQFB\_NOT\_DELIVERED.

La sortie qui s'affiche est la suivante:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

#### Exemple 2-Demande d'un message de réponse de trace-route

Générer et distribuer un message de trace-route dans la file d'attente cible

Dans cet exemple, l'application de route d'affichage IBM MQ se connecte au gestionnaire de files d'attente, QM1, et est utilisée pour générer et distribuer un message de route de trace à la file d'attente cible, TARGET.Q, sur le gestionnaire de files d'attente éloignées, QM2. L'option nécessaire est spécifiée de sorte que les informations d'activité soient accumulées dans le message de suivi. A l'arrivée dans la file d'attente cible, un message de réponse de suivi est demandé et le message de suivi est supprimé.

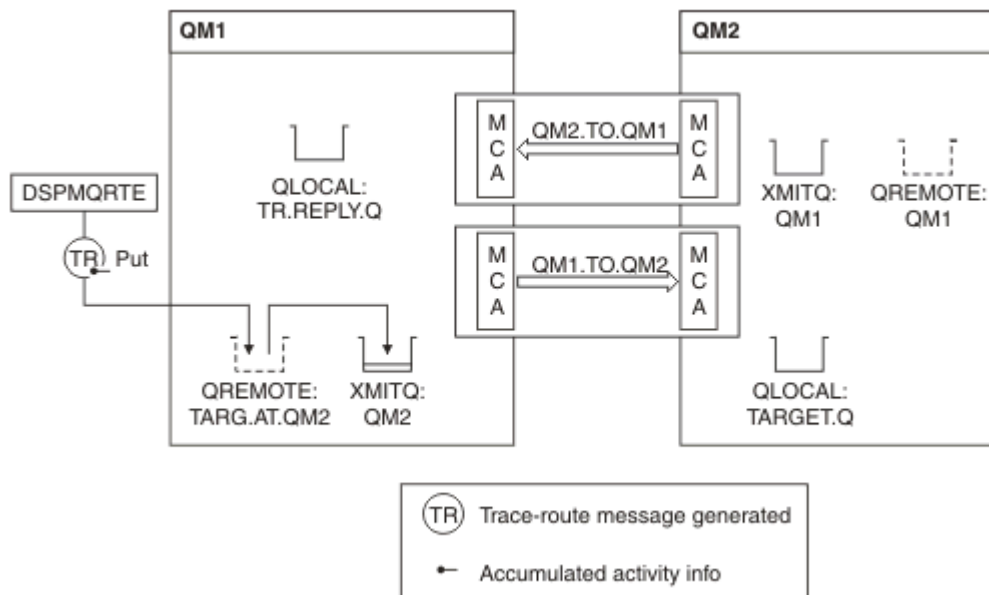


Figure 13. Demande d'un message de réponse de trace-route, diagramme 1

- L'attribut ROUTEREC de chaque gestionnaire de files d'attente (QM1 et QM2) est défini sur MSG.
- La commande suivante est émise:

```
dspmqrte -m QM1 -q TARG.AT.QM2 -rq TR.REPLY.Q -ac -ar -ro discard
```



QM1 est le nom du gestionnaire de files d'attente auquel l'application de route d'affichage IBM MQ se connecte, TARG.AT.QM2 est le nom de la file d'attente cible et ACTIV.REPLY.Q est le nom de la file d'attente à laquelle il est demandé que toutes les réponses au message de suivi soient envoyées. L'option -ac indique que les informations d'activité sont accumulées dans le message trace-route, l'option -ar indique que toutes les activités accumulées sont envoyées à la file d'attente de réponse spécifiée par l'option -rq (c'est-à-dire TR.REPLY.Q). L'option -ro indique que l'option de rapport MQRO\_DISCARD\_MSG est définie, ce qui signifie que les rapports d'activité ne sont pas générés dans cet exemple.

- DSPMQRTE accumule les informations d'activité dans le message de trace-route avant que le message ne soit placé sur la route cible. L'attribut de gestionnaire de files d'attente ROUTEREC sur QM1 ne doit pas être DISABLED pour que cela se produise.

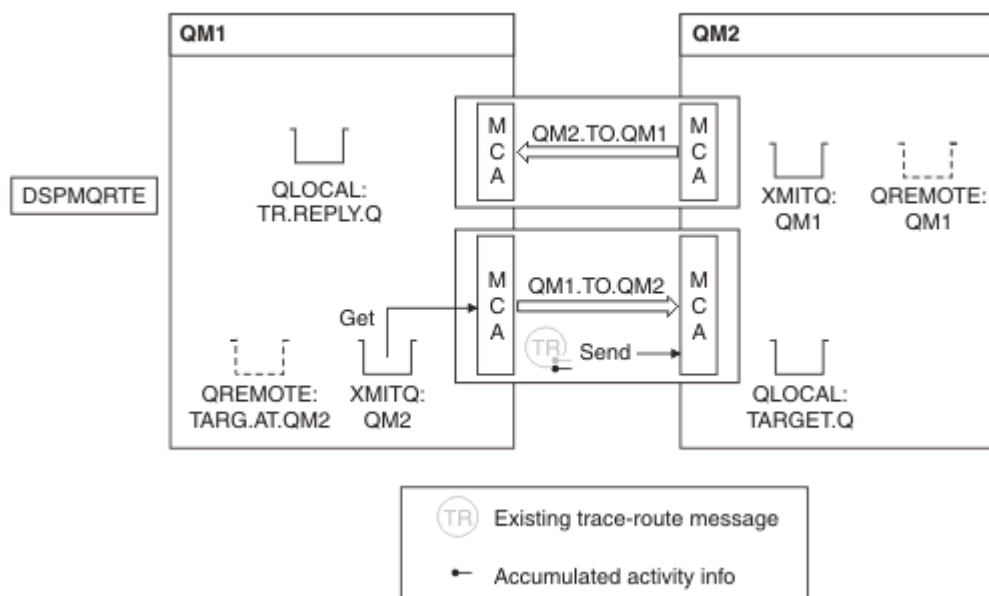


Figure 14. Demande d'un message de réponse de trace-route, diagramme 2

- Le message est un message d'acheminement de trace. Par conséquent, l'agent MCA émetteur commence à enregistrer des informations sur l'activité.
- L'attribut de gestionnaire de files d'attente ROUTEREC sur QM1 n'est pas DISABLED. Par conséquent, l'agent MCA accumule les informations d'activité dans le message, avant que celui-ci ne soit réacheminé vers le gestionnaire de files d'attente QM2.

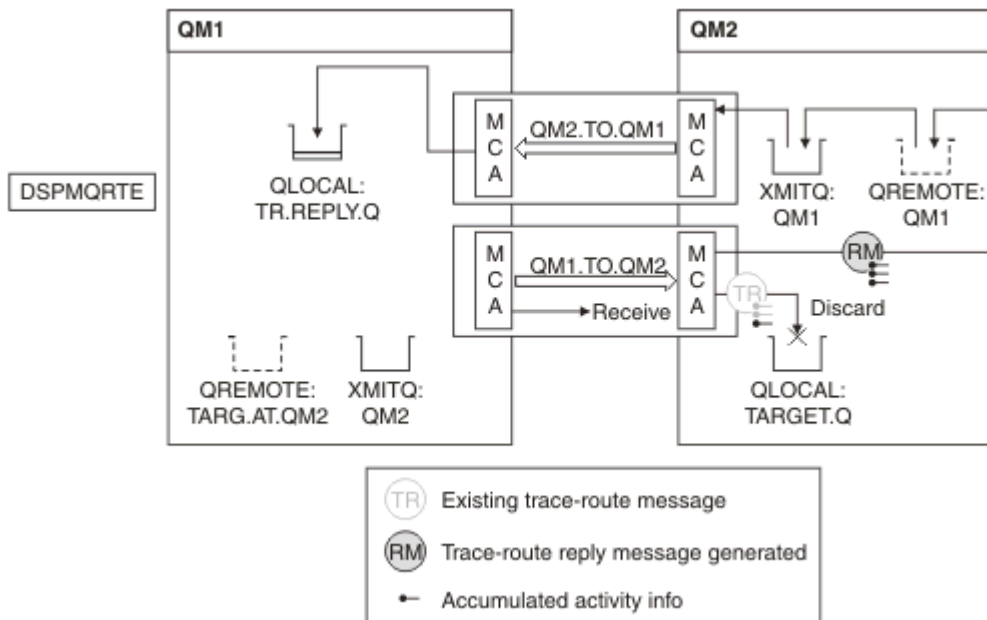


Figure 15. Demande d'un message de réponse de trace-route, diagramme 3

- Le message est un message d'acheminement de trace. Par conséquent, l'agent MCA récepteur commence à enregistrer des informations sur l'activité.
- L'attribut de gestionnaire de files d'attente ROUTEREC sur QM2 n'étant pas DISABLED, l'agent MCA accumule les informations dans le message.
- La file d'attente cible est une file d'attente locale. Par conséquent, le message est supprimé avec le retour d'informations MQFB\_NOT\_DELIVERED, conformément à la valeur du paramètre de distribution dans le groupe PCF TraceRoute .
- Il s'agit de la dernière activité qui aura lieu sur le message, et comme l'attribut de gestionnaire de files d'attente ROUTEREC sur QM1 n'est pas DISABLED, l'agent MCA génère un message de réponse de suivi conformément à la valeur d'accumulation. La valeur de ROUTEREC étant MSG, le message de réponse est placé dans la file d'attente de réponses. Le message de réponse contient toutes les informations d'activité cumulée provenant du message de trace-route.

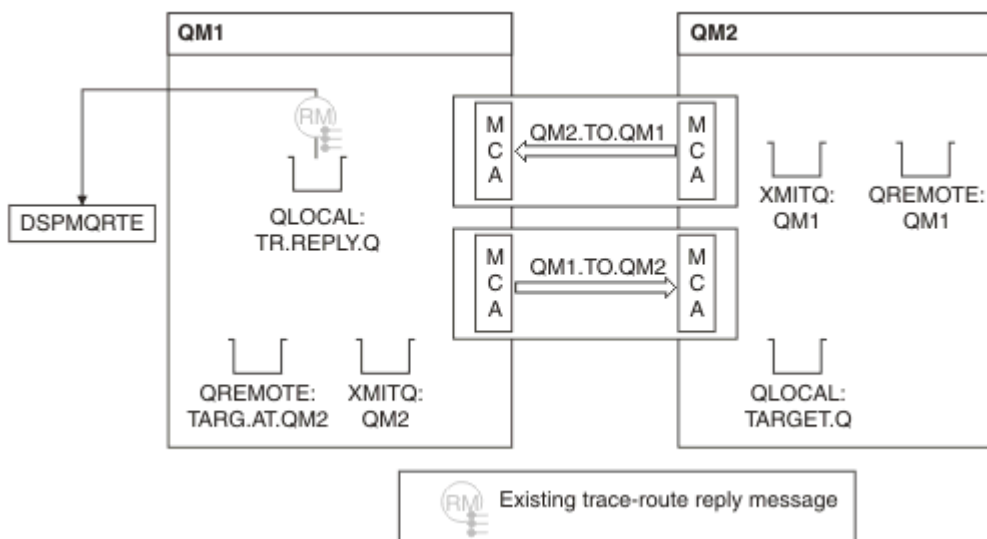


Figure 16. Demande d'un message de réponse de suivi, diagramme 4

- Dans le même temps, DSPMQRTE attend que le message de réponse de trace soit renvoyé dans la file d'attente de réponses. Lorsqu'il est renvoyé, DSPMQRTE analyse chaque activité qu'il contient et l'imprime. L'opération finale est une opération de suppression. La commande DSPMQRTE s'arrête une fois qu'elle a été imprimée.

La sortie qui s'affiche est la suivante:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
TR.REPLY.Q'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
AMQ8666: Queue 'QM2' on queue manager 'QM1'.
AMQ8666: Queue 'TARGET.Q' on queue manager 'QM2'.
AMQ8652: DSPMQRTE command has finished.
```

### Exemple 3-Distribution de rapports d'activité à la file d'attente système

Détectez le moment où les rapports d'activité sont distribués à des files d'attente autres que la file d'attente de réponse et utilisez l'application d'acheminement IBM MQ pour lire les rapports d'activité de l'autre file d'attente.

Cet exemple est identique à «Exemple 1-Demande de rapports d'activité», à la page 93, sauf que QM2 a désormais la valeur de l'attribut de gestion de file d'attente ACTIVREC définie sur QUEUE. Canal QM1.TO.QM2 doit avoir été redémarré pour que cette opération soit prise en compte.

Cet exemple montre comment détecter quand des rapports d'activité sont distribués à des files d'attente autres que la file d'attente de réponse. Une fois détectée, l'application d'affichage de route IBM MQ est utilisée pour lire les rapports d'activité d'une autre file d'attente.

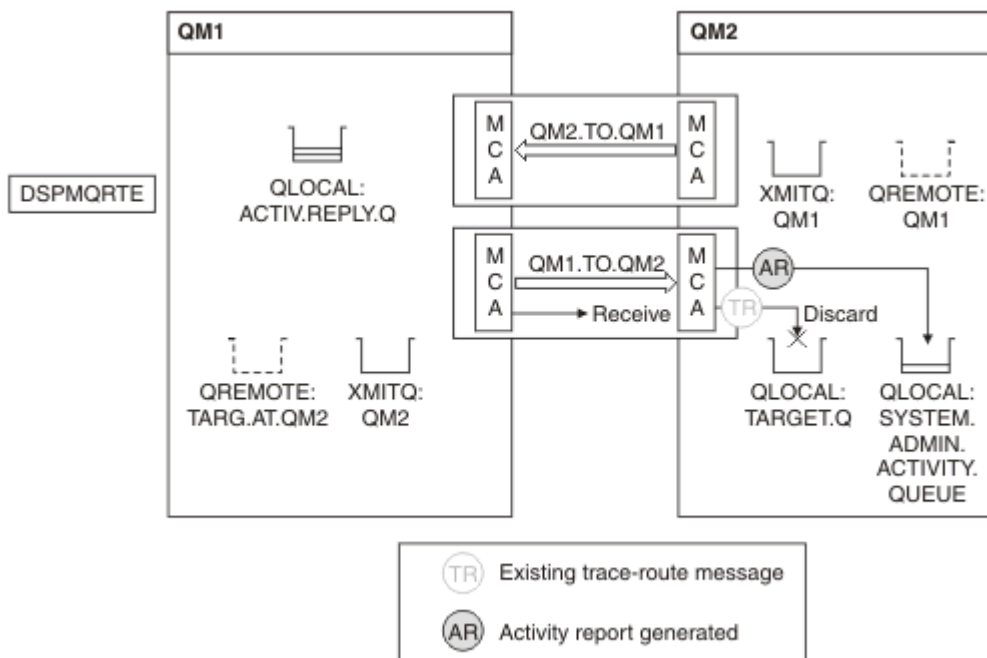


Figure 17. Distribution de rapports d'activité à la file d'attente système, diagramme 1

- Le message est un message d'acheminement de trace. Par conséquent, l'agent MCA récepteur commence à enregistrer des informations sur l'activité.
- La valeur de l'attribut de gestionnaire de files d'attente ACTIVREC sur QM2 est désormais QUEUE. Par conséquent, l'agent MCA génère un rapport d'activité, mais le place dans la file d'attente système (SYSTEM.ADMIN.ACTIVITY.QUEUE) et non dans la file d'attente de réponses (ACTIV.REPLY.Q).

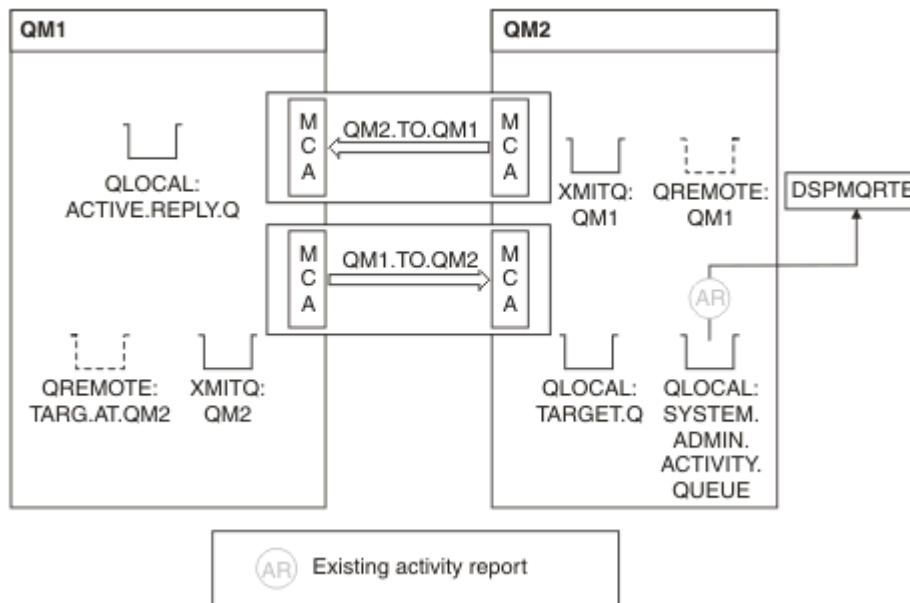


Figure 18. Distribution de rapports d'activité à la file d'attente système, diagramme 2

- Dans le même temps, DSPMQRTE attend l'arrivée des rapports d'activité sur `ACTIV.REPLY.Q`. Seulement deux arrivent. DSPMQRTE continue d'attendre 120 secondes car il semble que la route ne soit pas encore terminée.

La sortie qui s'affiche est la suivante:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2 -rq
ACTIV.REPLY.Q -v outline identifiers'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2', queue
manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
```

```
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'
```

```
Operation:
  OperationType: Put
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
```

```
-----
Activity:
  ApplName: 'cann\output\bin\runmqchl.EXE'
```

```
Operation:
  OperationType: Get
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001505'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001502'
```

```
EmbeddedMQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  QName: 'QM2'
  ResolvedQName: 'QM2'
```

```
Operation:
  OperationType: Send
```

```
Message:
```

```
MQMD:
  MsgId: X'414D51204C41524745512020202020A3C9154220001502'
  CorrelId: X'414D51204C41524745512020202020A3C9154220001503'
  QMgrName: 'QM1'
  RemoteQMgrName: 'QM2'
  ChannelName: 'QM1.TO.QM2'
  ChannelType: Sender
  XmitQName: 'QM2'
```

```
-----
AMQ8652: DSPMQRTE command has finished.
```

- La dernière opération observée par DSPMQRTE était un envoi. Par conséquent, le canal est en cours d'exécution. Nous devons maintenant déterminer la raison pour laquelle nous n'avons plus reçu de rapports d'activité du gestionnaire de files d'attente QM2 (comme indiqué dans le nom RemoteQMgr).
- Pour vérifier s'il existe des informations d'activité dans la file d'attente système, démarrez DSPMQRTE sur QM2 pour essayer de collecter d'autres rapports d'activité. Utilisez la commande suivante pour démarrer DSPMQRTE:

```
dspmqrte -m QM2 -q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C9154220001502 -v outline
```

où 414D51204C41524745512020202020A3C9154220001502 est le MsgId du message de trace-route qui a été inséré.

- DSPMQRTE effectue ensuite une séquence de MQGET à nouveau, en attendant les réponses dans la file d'attente d'activité du système liée au message de trace avec l'identificateur spécifié.

- DSPMQRTE obtient un autre rapport d'activité, qu'il affiche. DSPMQRTE détermine que les rapports d'activité précédents sont manquants et affiche un message le signalant. Cependant, nous connaissons déjà cette partie de la route.

La sortie qui s'affiche est la suivante:

```

AMQ8653: DSPMQRTE command started with options '-m QM2
-q SYSTEM.ADMIN.ACTIVITY.QUEUE
-i 414D51204C41524745512020202020A3C915420001502 -v outline'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----

Activity:
  Activity information unavailable.

-----

Activity:
  ApplName: 'cann\output\bin\AMQRMPPA.EXE'

  Operation:
    OperationType: Receive
    QMgrName: 'QM2'
    RemoteQMGrName: 'QM1'
    ChannelName: 'QM1.TO.QM2'
    ChannelType: Receiver

  Operation:
    OperationType: Discard
    QMgrName: 'QM2'
    QName: 'TARGET.Q'
    Feedback: NotDelivered

-----

AMQ8652: DSPMQRTE command has finished.

```

- Ce rapport d'activité indique que les informations de route sont désormais complètes. Aucun problème n'est survenu.
- Ce n'est pas parce que les informations de route ne sont pas disponibles ou parce que DSPMQRTE ne peut pas afficher toutes les routes, que le message n'a pas été distribué. Par exemple, les attributs de gestionnaire de files d'attente des différents gestionnaires de files d'attente peuvent être différents ou une file d'attente de réponses peut ne pas être définie pour récupérer la réponse.

#### *Exemple 4-Diagnostic d'un problème de canal*

Diagnostiquer un problème dans lequel le message de trace-route n'atteint pas la file d'attente cible

Dans cet exemple, l'application IBM MQ display route se connecte au gestionnaire de files d'attente, QM1, génère un message trace-route, puis tente de le distribuer à la file d'attente cible, TARGET.Q, sur le gestionnaire de files d'attente éloignées, QM2. Dans cet exemple, le message de trace-route n'atteint pas la file d'attente cible. Le rapport d'activités disponible est utilisé pour diagnostiquer le problème.

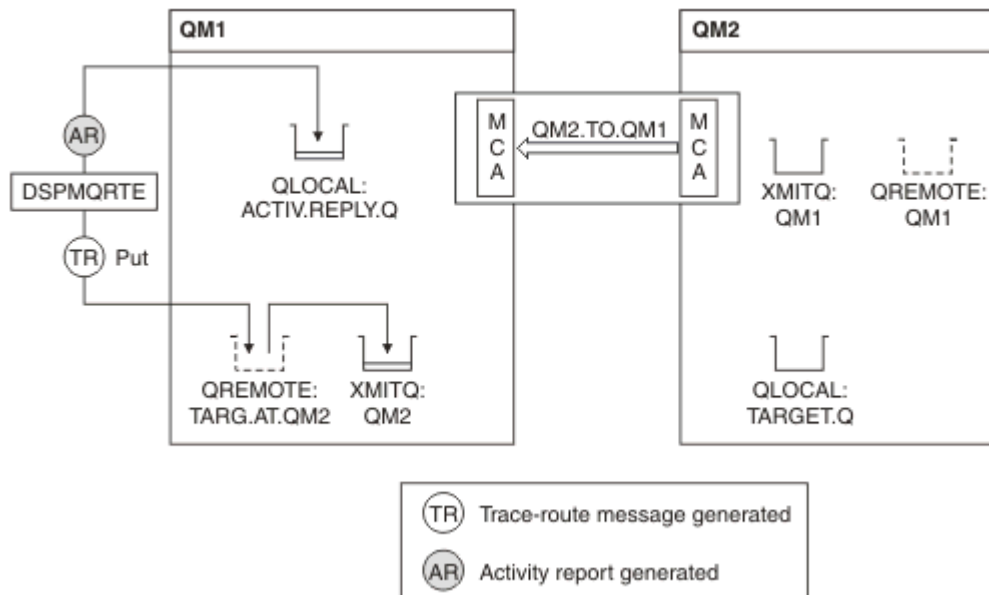


Figure 19. Diagnostic d'un problème de canal

- Dans cet exemple, le canal QM1.TO.QM2 n'est pas en cours d'exécution.
- DSPMQRTE insère un message de suivi (comme dans l'exemple 1) dans la file d'attente cible et génère un rapport d'activité.
- Il n'existe pas d'agent MCA pour extraire le message de la file d'attente de transmission (QM2). Il s'agit donc du seul rapport d'activité que DSPMQRTE extrait de la file d'attente de réponses. Cette fois, le fait que la route ne soit pas terminée indique un problème. L'administrateur peut utiliser la file d'attente de transmission trouvée dans ResolvedQName pour déterminer pourquoi la file d'attente de transmission n'est pas en cours de traitement.

La sortie qui s'affiche est la suivante:

```
AMQ8653: DSPMQRTE command started with options '-m QM1 -q TARG.AT.QM2
-rq ACTIV.REPLY.Q -v outline'.
AMQ8659: DSPMQRTE command successfully put a message on queue 'QM2',
queue manager 'QM1'.
AMQ8674: DSPMQRTE command is now waiting for information to display.
-----
Activity:
  ApplName: 'cann\output\bin\dspmqrte.exe'

Operation:
  OperationType: Put
  QMgrName: 'QM1'
  QName: 'TARG.AT.QM2'
  ResolvedQName: 'QM2'
  RemoteQName: 'TARGET.Q'
  RemoteQMgrName: 'QM2'
-----
AMQ8652: DSPMQRTE command has finished.
```

## Référence du rapport d'activité

Cette page permet d'obtenir une présentation du format de message du rapport d'activités. Les données de message du rapport d'activité contiennent les paramètres qui décrivent l'activité.

### Format du rapport d'activité

Les rapports d'activité sont des messages de rapport IBM MQ standard contenant un descripteur de message et des données de message. Les rapports d'activité sont des messages PCF générés par des

applications qui ont effectué une activité pour le compte d'un message lorsqu'il a été acheminé via un réseau de gestionnaires de files d'attente.

Les rapports d'activité contiennent les informations suivantes:

**Un descripteur de message**

Une structure MQMD

**Données de message**

Comprend les éléments suivants:

- En-tête PCF imbriqué (MQEPH).
- Données de message du rapport d'activité.

Les données de message de rapport d'activité sont constituées du groupe PCF *Activity* et, si elles sont générées pour un message de trace, du groupe PCF *TraceRoute* .

Le [Tableau 19](#), à la page 105 présente la structure de ces rapports, y compris les paramètres renvoyés uniquement sous certaines conditions.



Tableau 19. Format du rapport d'activité

Structure MQMD	Structure MQEPH d'en-tête PCF intégrée	Données de message du rapport d'activité
Identificateur de structure Version de la structure Options de rapport Type de message Délai d'expiration Commentaires Codage CCSID Format de message Priorité Persistance Identificateur de message Identificateur de corrélation Nombre d'annulations File de réponses Gestionnaire de file de réponses ID utilisateur Jeton de comptabilité Données sur l'identité de l'application Type d'application Nom d'application Date d'insertion Heure d'insertion Données sur l'origine de l'application Identificateur de groupe Numéro de séquence du message Décalage Indicateurs de message Longueur d'origine	Identificateur de structure Version de la structure Longueur de la structure Codage CCSID Format de message Indicateurs En-tête PCF (MQCFH) Type de structure Longueur de la structure Version de la structure Identificateur de commande Numéro de séquence du message Options de contrôle Code de fin d'exécution Code raison Nombre de paramètres	Activité Nom de l'application d'activité Type d'application d'activité Description de l'activité Opération Type d'opération Date de l'opération Heure de l'opération Message LONGUEUR DE MESSAGE MQMD <sup>8</sup> EmbeddedMQMD Nom gest. de files Nom du groupe de partage de files d'attente Nom de la file d'attente <sup>1</sup> <sup>2</sup> <sup>3</sup> <sup>7</sup> Nom de file d'attente résolu <sup>1</sup> <sup>3</sup> <sup>7</sup> Nom de la file d'attente éloignée <sup>3</sup> <sup>7</sup> Nom du gestionnaire de files d'attente éloignées <sup>2</sup> <sup>3</sup> <sup>4</sup> <sup>5</sup> <sup>7</sup> Niveau d'abonnement <sup>9</sup> Identificateur d'abonnement <sup>9</sup> Commentaires en retour <sup>2</sup> <sup>10</sup> Nom de canal <sup>4</sup> <sup>5</sup> Type de canal <sup>4</sup> <sup>5</sup> Nom de la file d'attente de transmission <sup>5</sup> TraceRoute <sup>6</sup> Détails Activités enregistrées Activités non enregistrées Nombre de discontinuités Activités max Cumuler Délivrer

**Remarques :**

1. Renvoyé pour les opérations d'extraction et de navigation.
2. Renvoyé pour les opérations de suppression.

3. Renvoyé pour les opérations d'insertion, de réponse d'insertion et de rapport d'insertion.
4. Renvoyé pour les opérations de réception.
5. Renvoyé pour les opérations d'envoi.
6. Renvoyé pour les messages de trace-route.
7. Non renvoyé pour les opérations d'insertion dans une rubrique, contenue dans les activités de publication.
8. Non renvoyé pour les opérations de publication exclues. Pour les opérations de publication et de suppression de publication, renvoyé contenant un sous-ensemble de paramètres.
9. Renvoyé pour les opérations de publication, de suppression de publication et de publication exclue.
10. Renvoyé pour les opérations de publication et de publication exclues supprimées.

### **Rapport d'activité MQMD (descripteur de message)**

Cette page permet d'afficher les valeurs contenues dans la structure MQMD d'un rapport d'activité

#### **StrucId**

Identificateur de structure:

##### **Type de données**

MQCHAR4

##### **Valeur**

MQMD\_STRUC\_ID.

#### **Version**

Numéro de version de structure

##### **Type de données**

MLONG

##### **Valeurs**

Copié à partir du descripteur de message d'origine. Les valeurs possibles sont les suivantes :

##### **MQMD\_VERSION\_1**

Structure de descripteur de message Version-1 , prise en charge dans tous les environnements.

##### **MQMD\_VERSION\_2**

Structure de descripteur de message Version-2 , prise en charge dans les environnements suivants:

-  AIX
-  IBM i
-  Linux
-  Windows
-  z/OS

et tous les IBM MQ MQI clients connectés à ces systèmes.

#### **Report**

Options pour les autres messages de rapport

##### **Type de données**

MLONG

##### **Valeur**

Si MQRO\_PASS\_DISCARD\_AND\_EXPIRATION ou MQRO\_DISCARD\_MSG a été spécifié dans la zone *Rapport* du descripteur de message d'origine:

##### **Carte MQRO\_DISCARD**

Le rapport est supprimé s'il ne peut pas être distribué à la file d'attente de destination.

Sinon :

**MQRO\_AUCUN**

Aucun rapport n'est requis.

**MsgType**

Indique le type de message

**Type de données**

MQLONG

**Valeur**

MQMT\_REPORT

**Expiry**

Durée de vie des messages de rapport

**Type de données**

MQLONG

**Valeur**

Si la zone *Rapport* du descripteur de message d'origine est définie sur MQRO\_PASS\_DISCARD\_AND\_EXPIRATION, le délai d'expiration restant à partir du message d'origine est utilisé.

Sinon :

**MQEI\_UNLIMITED**

Le rapport n'a pas de délai d'expiration.

**Feedback**

Description : Retour d'informations ou code anomalie.

Type de données : MQLONG:

Valeur : **ACTIVITE MQFB\_MQ**  
Rapport d'activité.

**Encoding**

Description : Codage numérique des données de message de rapport.

Type de données : MQLONG:

Valeur : MQENC\_NATIVE.

**CodedCharSetId**

Description : Identificateur de jeu de caractères des données de message de rapport.

Type de données : MQLONG:

Valeur : Définisseau besoin.

**Format**

Description : Nom de format des données de message de rapport

Type de données : MQCHAR8.

Valeur : **MQFMT\_EMBEDDED\_PCF**  
Message PCF imbriqué.

**Priority**

Description : Indique la priorité des messages.

Type de données : MQLONG:  
Valeur : Copié à partir du descripteur de message d'origine.

### ***Persistence***

Description : Signaler la persistance des messages.  
Type de données : MQLONG:  
Valeur : Copié à partir du descripteur de message d'origine.

### ***MsgId***

Description : Identificateur de message.  
Type de données : MQBYTE24.  
Valeurs : Si la zone *Rapport* du descripteur de message d'origine est spécifiée en tant que MQRO\_PASS\_MSG\_ID, l'identificateur du message d'origine est utilisé.  
Sinon, une valeur unique sera générée par le gestionnaire de files d'attente.

### ***CorrelId***

Description : Identificateur de corrélation.  
Type de données : MQBYTE24.  
Valeur : Si la zone *Rapport* du descripteur de message d'origine est spécifiée en tant que MQRO\_PASS\_CORREL\_ID, l'identificateur de corrélation du message d'origine est utilisé.  
Sinon, l'identificateur de message est copié à partir du message d'origine.

### ***BackoutCount***

Description : Compteur d'annulation.  
Type de données : MQLONG:  
Valeur : 0.

### ***ReplyToQ***

Description : Nom de la file d'attente de réponses.  
Type de données : MQCHAR48.  
Valeurs : Non renseigné.

### ***ReplyToQMgr***

Description : Nom du gestionnaire de files d'attente de réponses.  
Type de données : MQCHAR48.  
Valeur : Nom du gestionnaire de files d'attente qui a généré le message de rapport.

### ***UserIdentifier***

Description : Identificateur utilisateur de l'application qui a généré le message de rapport.  
Type de données : MQCHAR12.  
Valeur : Copié à partir du descripteur de message d'origine.

### **AccountingToken**

Description : Jeton de comptabilité qui permet à une application de facturer le travail effectué à la suite du message.

Type de données : MQBYTE32.

Valeur : Copié à partir du descripteur de message d'origine.

### **ApplIdentityData**

Description : Données d'application relatives à l'identité.

Type de données : MQCHAR32.

Valeurs : Copié à partir du descripteur de message d'origine.

### **PutApplType**

Description : Type d'application qui a inséré le message de rapport.

Type de données : MQLONG:

Valeur : **MQAT\_QMGR**  
Message généré par le gestionnaire de files d'attente.

### **PutApplName**

Description : Nom de l'application qui a inséré le message de rapport.

Type de données : MQCHAR28.

Valeur : Les 28 premiers octets du nom du gestionnaire de files d'attente ou le nom de l'agent MCA qui a généré le message de rapport.

### **PutDate**

Description : Date à laquelle le message a été inséré.

Type de données : MQCHAR8.

Valeur : Généré par le gestionnaire de files d'attente.

### **PutTime**

Description : Heure à laquelle le message a été inséré.

Type de données : MQCHAR8.

Valeur : Généré par le gestionnaire de files d'attente.

### **ApplOriginData**

Description : Données d'application relatives à l'origine.

Type de données : MQCHAR4.

Valeur : Non renseigné.

Si *Version* est MQMD\_VERSION\_2, les zones supplémentaires suivantes sont disponibles:

### **GroupId**

Description : Identifie le groupe de messages ou le message logique auquel appartient le message physique.

Type de données : MQBYTE24.

Valeur : Copié à partir du descripteur de message d'origine.

### ***MsgSeqNumber***

Description : Numéro de séquence du message logique dans le groupe.

Type de données : MQLONG:

Valeur : Copié à partir du descripteur de message d'origine.

### ***Offset***

Description : Décalage des données dans le message physique à partir du début du message logique.

Type de données : MQLONG:

Valeur : Copié à partir du descripteur de message d'origine.

### ***MsgFlags***

Description : Indicateurs de message qui spécifient les attributs du message ou contrôlent son traitement.

Type de données : MQLONG:

Valeur : Copié à partir du descripteur de message d'origine.

### ***OriginalLength***

Description : Longueur du message d'origine.

Type de données : MQLONG:

Valeur : Copié à partir du descripteur de message d'origine.

### ***Rapport d'activité MQEPH (en-tête PCF imbriqué)***

Cette page permet d'afficher les valeurs contenues dans la structure MQEPH pour un rapport d'activité.

La structure MQEPH contient une description des informations PCF qui accompagnent les données de message d'un rapport d'activité et des données de message d'application qui le suivent.

Pour un rapport d'activité, la structure MQEPH contient les valeurs suivantes:

#### ***StrucId***

Description : Identificateur de structure.

Type de données : MQCHAR4.

Valeur : MQEPH\_STRUC\_ID.

#### ***Version***

Description : Numéro de version de la structure.

Type de données : MQLONG:

Valeurs : MQEPH\_VERSION\_1.

#### ***StrucLength***

Description : Longueur de la structure.

Type de données : MQLONG:

Valeur : Longueur totale de la structure, y compris les structures de paramètre PCF qui la suivent.

### **Encoding**

Description : Codage numérique des données de message qui suivent la dernière structure de paramètre PCF.

Type de données : MQLONG:

Valeur : Si des données des données de message d'application d'origine sont incluses dans le message de rapport, la valeur est copiée à partir de la zone *Codage* du descripteur de message d'origine.

Sinon, 0.

### **CodedCharSetId**

Description : Identificateur de jeu de caractères des données de message qui suivent la dernière structure de paramètre PCF.

Type de données : MQLONG:

Valeur : Si des données des données de message d'application d'origine sont incluses dans le message de rapport, la valeur est copiée à partir de la zone *CodedCharSetId* du descripteur de message d'origine.

Sinon, MQCCSI\_UNDEFINED.

### **Format**

Description : Nom de format des données de message qui suivent la dernière structure de paramètre PCF.

Type de données : MQCHAR8.

Valeur : Si des données des données de message d'application d'origine sont incluses dans le message de rapport, la valeur est copiée à partir de la zone *Format* du descripteur de message d'origine.

Sinon, MQFMT\_NONE.

### **Flags**

Description : Indicateurs spécifiant les attributs de la structure ou contrôlant son traitement.

Type de données : MQLONG:

Valeur : **MQEPH\_CCSID\_IMBRIQUÉ**

Indique que le jeu de caractères des paramètres contenant des données de type caractères est spécifié individuellement dans la zone *CodedCharSetId* de chaque structure.

### **PCFHeader**

Description : En-tête de format de commande programmable

Type de données : MQCFH.

Valeur : Voir [«Rapport d'activité MQCFH \(en-tête PCF\)»](#), à la page 111.

### **Rapport d'activité MQCFH (en-tête PCF)**

Cette page permet d'afficher les valeurs PCF contenues dans la structure MQCFH d'un rapport d'activité.

Pour un rapport d'activité, la structure MQCFH contient les valeurs suivantes:

### **Type**

Description : Type de structure qui identifie le contenu du message de rapport.  
Type de données : MQLONG:  
Valeur : **RAPPORT MQCF**  
Le message est un rapport.

### **StrucLength**

Description : Longueur de la structure.  
Type de données : MQLONG:  
Valeur : **LONGUEUR\_STRUC\_MQCFH**  
Longueur en octets de la structure MQCFH.

### **Version**

Description : Numéro de version de la structure.  
Type de données : MQLONG:  
Valeurs : MQCFH\_VERSION\_3

### **Command**

Description : Identificateur de commande. Identifie la catégorie du message.  
Type de données : MQLONG:  
Valeurs : **MQCMD\_ACTIVITY\_MSG**  
Activité des messages.

### **MsgSeqNumber**

Description : Numéro de séquence du message. Il s'agit du numéro de séquence du message dans un groupe de messages associés.  
Type de données : MQLONG:  
Valeurs : 1.

### **Control**

Description : Options de contrôle.  
Type de données : MQLONG:  
Valeurs : MQCFC\_LAST.

### **CompCode**

Description : Code achèvement.  
Type de données : MQLONG:  
Valeurs : MQCC\_OK.

### **Reason**

Description : Code raison qualifiant le code achèvement.



Type de données : MQLONG:  
Valeurs : MQRC\_NONE:

### **ParameterCount**

Description : Nombre de structures de paramètres. Il s'agit du nombre de structures de paramètres qui suivent la structure MQCFH. Une structure de groupe (MQCFGR) et ses structures de paramètres incluses sont comptées comme une seule structure.

Type de données : MQLONG:  
Valeurs : 1 ou supérieur.

### **Données de message du rapport d'activité**

Cette page permet d'afficher les paramètres contenus par le groupe PCF *Activité* dans un message de rapport d'activité. Certains paramètres sont renvoyés uniquement lorsque des opérations spécifiques ont été effectuées.

Les données de message de rapport d'activité sont constituées du groupe PCF *Activité* et, si elles sont générées pour un message de trace, du groupe PCF *TraceRoute*. Le groupe PCF *Activité* est détaillé dans cette rubrique.

Certains paramètres, qui sont décrits comme Données de message de rapport d'activité spécifique à l'opération, sont renvoyés uniquement lorsque des opérations spécifiques ont été effectuées.

Pour un rapport d'activité, les données de message du rapport d'activité contiennent les paramètres suivants:

### **Activity**

Description : Paramètres groupés décrivant l'activité.

Identificateur : MQGACF\_ACTIVITY.

Type de données : MQCFGR.

Inclus dans le groupe PCF: Néant.

Paramètres du groupe PCF: *ActivityApplName*  
*ActivityApplType*  
*ActivityDescription*  
*Operation*  
*TraceRoute*

Renvoyé: Toujours.

### **ActivityApplName**

Description : Nom de l'application qui a effectué l'activité.

Identificateur : MQCACF\_APPL\_NAME.

Type de données : MQCFST.

Inclus dans le groupe PCF: *Activité*.

Longueur maximale : MQ\_APPL\_NAME\_LENGTH.

Renvoyé: Toujours.

### **ActivityApplType**

Description :	Type d'application qui a effectué l'activité.
Identificateur :	MQIA_APPL_TYPE.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Activité.</i>
Renvoyé:	Toujours.

### **ActivityDescription**

Description :	Description de l'activité effectuée par l'application.
Identificateur :	MQCACF_ACTIVITY_DESCRIPTION.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Activité.</i>
Longueur maximale :	64
Renvoyé:	Toujours.

### **Operation**

Description :	Paramètres groupés décrivant une opération de l'activité.
Identificateur :	MQGACF_OPERATION.
Type de données :	MQCFGR.
Inclus dans le groupe PCF:	<i>Activité.</i>
Paramètres du groupe PCF:	<i>OperationType OperationDate OperationTime Message QMgrName QSGName</i>

**Remarque :** Des paramètres supplémentaires sont renvoyés dans ce groupe en fonction du type d'opération. Ces paramètres supplémentaires sont décrits comme Données de message de rapport d'activité spécifique à l'opération.

Renvoyé:	Un groupe PCF <i>Opération</i> par opération dans l'activité.
----------	---

### **OperationType**

Description :	Type d'opération effectuée.
Identificateur :	MQIACF_TYPE_OPÉRATION.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Opération.</i>
Valeurs :	MQOPER_ *.
Renvoyé:	Toujours.

### **OperationDate**

Description : Date à laquelle l'opération a été effectuée.  
Identificateur : MQCACF\_OPERATION\_DATE.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *Opération.*  
Longueur maximale : MQ\_DATE\_LENGTH.  
Renvoyé: Toujours.

### **OperationTime**

Description : Heure à laquelle l'opération a été effectuée.  
Identificateur : MQCACF\_HEURE.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *Opération.*  
Longueur maximale : MQ\_TIME\_LENGTH.  
Renvoyé: Toujours.

### **Message**

Description : Paramètres groupés décrivant le message à l'origine de l'activité.  
Identificateur : MQGACF\_MESSAGE.  
Type de données : MQCFGR.  
Inclus dans le groupe PCF: *Opération.*  
Paramètres du groupe: *MsgLength*  
*MQMD*  
*EmbeddedMQMD*  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### **MsgLength**

Description : Longueur du message à l'origine de l'activité, avant que celle-ci ne se produise.  
Identificateur : MQIACF\_MSG\_LENGTH.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *Message.*  
Renvoyé: Toujours.

### **MQMD**

Description : Paramètres groupés liés au descripteur de message du message à l'origine de l'activité.  
Identificateur : MQGACF\_MQMD.

Type de données : MQCFGR.

Inclus dans le groupe PCF: *Message.*

Paramètres du groupe:

- StrucId*
- Version*
- Report*
- MsgType*
- Expiry*
- Feedback*
- Encoding*
- CodedCharSetId*
- Format*
- Priority*
- Persistence*
- MsgId*
- CorrelId*
- BackoutCount*
- ReplyToQ*
- ReplyToQMgr*
- UserIdentifier*
- AccountingToken*
- ApplIdentityData*
- PutApplType*
- PutApplName*
- PutDate*
- PutTime*
- ApplOriginData*
- GroupId*
- MsgSeqNumber*
- Offset*
- MsgFlags*
- OriginalLength*

Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### ***EmbeddedMQMD***

Description : Paramètres regroupés décrivant le descripteur de message imbriqué dans un message dans une file d'attente de transmission.

Identificateur : MQGACF\_EMBEDDED\_MQMD.

Type de données : MQCFGR.

Inclus dans le groupe PCF: *Message.*

Paramètres du groupe:	<i>StrucId</i> <i>Version</i> <i>Report</i> <i>MsgType</i> <i>Expiry</i> <i>Feedback</i> <i>Encoding</i> <i>CodedCharSetId</i> <i>Format</i> <i>Priority</i> <i>Persistence</i> <i>MsgId</i> <i>CorrelId</i> <i>BackoutCount</i> <i>ReplyToQ</i> <i>ReplyToQMgr</i> <i>UserIdentifier</i> <i>AccountingToken</i> <i>ApplIdentityData</i> <i>PutApplType</i> <i>PutApplName</i> <i>PutDate</i> <i>PutTime</i> <i>ApplOriginData</i> <i>GroupId</i> <i>MsgSeqNumber</i> <i>Offset</i> <i>MsgFlags</i> <i>OriginalLength</i>
-----------------------	---

Renvoyé: Pour les opérations d'extraction où la file d'attente se résout en file d'attente de transmission.

### **StrucId**

Description : Identificateur de structure

Identificateur : MQCACF\_STRUC\_ID.

Type de données : MQCFST.

Inclus dans le groupe PCF: MQMD ou *EmbeddedMQMD*.

Longueur maximale : 4.

Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Version**

Description : Numéro de version de la structure.

Identificateur : MQIACF\_VERSION.

Type de données : MQCFIN.

Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Report**

Description : Options des messages de rapport.  
Identificateur : RAPPOR MQIACF\_REPORT.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **MsgType**

Description : Indique le type de message.  
Identificateur : MQIACF\_MSG\_TYPE.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Expiry**

Description : Durée de vie du message.  
Identificateur : MQIACF\_EXPIRATION.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Feedback**

Description : Retour d'informations ou code anomalie.  
Identificateur : MQIACF\_FEEDBACK.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Encoding**

Description : Codage numérique des données de message.  
Identificateur : MQIACF\_ENCODING.

Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **CodedCharSetId**

Description : Identificateur de jeu de caractères des données d'un message.  
Identificateur : MQIA\_CODED\_CHAR\_SET\_ID.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Format**

Description : Nom de format des données de message  
Identificateur : MQCACH\_FORMAT\_NAME.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_FORMAT\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### **Priority**

Description : Priorité du message.  
Identificateur : MQIACF\_PRIORITY.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### **Persistence**

Description : Persistence des messages.  
Identificateur : MQIACF\_PERSISTENCE.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### **MsgId**

Description : Identificateur de message.

Identificateur : MQBACF\_MSG\_ID.  
Type de données : MQCFBS.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_MSG\_ID\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### ***CorrelId***

Description : Identificateur de corrélation.  
Identificateur : MQBACF\_CORREL\_ID.  
Type de données : MQCFBS.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_CORREL\_ID\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### ***BackoutCount***

Description : Compteur d'annulation.  
Identificateur : MQIACF\_BACKOUT\_COUNT.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans *MQMD* pour les opérations de publication et de suppression.

### ***ReplyToQ***

Description : Nom de la file d'attente de réponses.  
Identificateur : MQCACF\_REPLY\_TO\_Q.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_Q\_NAME\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans *MQMD* pour les opérations de publication et de suppression.

### ***ReplyToMgr***

Description : Nom du gestionnaire de files d'attente de réponses.  
Identificateur : MQCACF\_REPLY\_TO\_Q\_MGR.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.



Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### ***UserIdentifier***

Description : Identificateur utilisateur de l'application à l'origine du message.  
Identificateur : MQCACF\_IDENTIFICATEUR\_UTILISATEUR.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_USER\_ID\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### ***AccountingToken***

Description : Jeton de comptabilité qui permet à une application de facturer le travail effectué à la suite du message.  
Identificateur : MQBACF\_ACCOUNTING\_TOKEN.  
Type de données : MQCFBS.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_ACCOUNTING\_TOKEN\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### ***ApplIdentityData***

Description : Données d'application relatives à l'identité.  
Identificateur : MQCACF\_APPL\_IDENTITY\_DATA.  
Type de données : MQCFST.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Longueur maximale : MQ\_APPL\_IDENTITY\_DATA\_LENGTH.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues.

### ***PutApplType***

Description : Type d'application qui a inséré le message.  
Identificateur : MQIA\_APPL\_TYPE.  
Type de données : MQCFIN.  
Inclus dans le groupe PCF: *MQMD* ou *EmbeddedMQMD*.  
Renvoyé: Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **PutApplName**

Description :	Nom de l'application qui a inséré le message.
Identificateur :	MQCACF_APPL_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Longueur maximale :	MQ_APPL_NAME_LENGTH.
Renvoyé:	Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **PutDate**

Description :	Date à laquelle le message a été inséré.
Identificateur :	MQCACF_PUT_DATE:
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Longueur maximale :	MQ_PUT_DATE_LENGTH.
Renvoyé:	Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **PutTime**

Description :	Heure à laquelle le message a été inséré.
Identificateur :	MQCACF_PUT_TIME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Longueur maximale :	MQ_PUT_TIME_LENGTH.
Renvoyé:	Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **ApplOriginData**

Description :	Données d'application relatives à l'origine.
Identificateur :	MQCACF_APPL_ORIGIN_DATA.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Longueur maximale :	MQ_APPL_ORIGIN_DATA_LENGTH.
Renvoyé:	Toujours, sauf pour les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **GroupId**

Description :	Identifie le groupe de messages ou le message logique auquel appartient le message physique.
Identificateur :	ID_GROUPE_MQBACF.
Type de données :	MQCFBS.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Longueur maximale :	MQ_GROUP_ID_LENGTH.
Renvoyé:	Si la <i>version</i> est spécifiée en tant que MQMD_VERSION_2. Non renvoyé dans les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **MsgSeqNumber**

Description :	Numéro de séquence du message logique dans le groupe.
Identificateur :	MQIACH_MSG_SEQUENCE_NUMBER.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Renvoyé:	Si <i>Version</i> est spécifié en tant que MQMD_VERSION_2. Non renvoyé dans les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **Offset**

Description :	Décalage des données dans le message physique à partir du début du message logique.
Identificateur :	MQIACF_OFFSET.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Renvoyé:	Si <i>Version</i> est spécifié en tant que MQMD_VERSION_2. Non renvoyé dans les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **MsgFlags**

Description :	Indicateurs de message qui spécifient les attributs du message ou contrôlent son traitement.
Identificateur :	MQIACF_MSG_FLAGS.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Renvoyé:	Si <i>Version</i> est spécifié en tant que MQMD_VERSION_2. Non renvoyé dans les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **OriginalLength**

Description :	Longueur du message d'origine.
Identificateur :	MQIACF_ORIGINAL_LENGTH.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>MQMD</i> ou <i>EmbeddedMQMD</i> .
Renvoyé:	Si <i>Version</i> est spécifié en tant que MQMD_VERSION_2. Non renvoyé dans les opérations de publication exclues et dans MQMD pour les opérations de publication et de suppression.

### **QMgrName**

Description :	Nom du gestionnaire de files d'attente dans lequel l'activité a été effectuée.
Identificateur :	MQCA_Q_MGR_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération</i> .
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH
Renvoyé:	Toujours.

### **QSGName**

Description :	Nom du groupe de partage de files d'attente auquel appartient le gestionnaire de files d'attente dans lequel l'activité a été effectuée.
Identificateur :	MQCA_QSG_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération</i> .
Longueur maximale :	LONGUEUR_NOM_QSG_MQ_
Renvoyé:	Si l'activité a été effectuée sur un gestionnaire de files d'attente IBM MQ for z/OS .

### **TraceRoute**

Description :	Paramètres groupés spécifiant les attributs du message de trace-route.
Identificateur :	MQGACF_TRACE_ROUTE.
Type de données :	MQCFGR.
Contenu dans le groupe PCF:	<i>Activité</i> .

Paramètres du groupe:	<i>Detail</i> <i>RecordedActivities</i> <i>UnrecordedActivities</i> <i>DiscontinuityCount</i> <i>MaxActivities</i> <i>Accumulate</i> <i>Forward</i> <i>Deliver</i>
-----------------------	---

Renvoyé: Indique si l'activité a été effectuée pour le compte du message de suivi.

Les valeurs des paramètres du groupe PCF *TraceRoute* sont celles du message trace-route au moment de la génération du rapport d'activité.

### **Données de message du rapport d'activité spécifique à l'opération**

Cette page permet d'afficher les paramètres PCF supplémentaires qui peuvent être renvoyés dans le groupe PCF *Opération* dans un rapport d'activité, en fonction de la valeur du paramètre *OperationType*

Les paramètres supplémentaires varient en fonction des types d'opération suivants:

#### *Obtenir / Parcourir (MQOPER\_GET/MQOPER\_BROWSE)*

Les paramètres de données de message de rapport d'activité supplémentaires qui sont renvoyés dans le groupe PCF *Opération* pour le type d'opération Get / Browse (MQOPER\_GET/MQOPER\_BROWSE) (un message sur une file d'attente a été obtenu ou consulté).

#### **QName**

Description :	Nom de la file d'attente qui a été ouverte.
Identificateur :	MQCA_Q_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	LONGUEUR_NOM_Q_MQ_
Renvoyé:	Toujours.

#### **ResolvedQName**

Description :	Nom de la file d'attente ouverte.
Identificateur :	MQCACF_RESOLVED_Q_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	LONGUEUR_NOM_Q_MQ_
Renvoyé:	Toujours.

#### *Suppression (MQOPER\_DISCARD)*

Les paramètres de données de message de rapport d'activité supplémentaires qui sont renvoyés dans le groupe PCF *Opération* pour le type d'opération Discard (MQOPER\_DISCARD) (un message a été supprimé).

### **Feedback**

Description :	Motif de la suppression du message.
Identificateur :	MQIACF_FEEDBACK.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Toujours.

### **QName**

Description :	Nom de la file d'attente qui a été ouverte.
Identificateur :	MQCA_Q_NAME.
Type de données :	MQCFST.
Longueur maximale :	LONGUEUR_NOM_Q_MQ_
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Si le message a été supprimé parce qu'il n'a pas été inséré dans une file d'attente.

### **RemoteQMgrName**

Description :	Nom du gestionnaire de files d'attente auquel le message a été destiné.
Identificateur :	MQCA_REMOTE_Q_MGR_NAME.
Type de données :	MQCFST.
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Si la valeur de <i>Feedback</i> est MQFB_NOT_TRANSMIT.

*Publication / Publication supprimée / Publication exclue (MQOPER\_PUBLISH/MQOPER\_DISCARDED\_PUBLISH/MQOPER\_EXCLUDED\_PUBLISH)*

Les paramètres de données de message de rapport d'activité supplémentaires qui sont renvoyés dans le groupe PCF *Opération* pour le type d'opération Publier / Supprimer / Publier exclu (MQOPER\_PUBLISH/MQOPER\_DISCARDED\_PUBLISH/MQOPER\_EXCLUDED\_PUBLISH) (un message de publication / abonnement a été distribué, supprimé ou exclu).

### **SubId**

Description :	Identificateur de l'abonnement.
Identificateur :	MQBACF_SUB_ID.
Type de données :	MQCFBS.
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Toujours.

### **SubLevel**

Description :	Niveau d'abonnement.
Identificateur :	MQIACF_SUB_LEVEL.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Toujours.

### **Feedback**

Description :	Motif de la suppression du message.
Identificateur :	MQIACF_FEEDBACK.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Si le message a été supprimé car il n'a pas été distribué à un abonné, ou si le message n'a pas été distribué car l'abonné a été exclu.

L'opération de publication MQOPER\_PUBLISH fournit des informations sur un message distribué à un abonné particulier. Cette opération décrit les éléments du message en cours qui ont pu être modifiés par rapport au message décrit dans l'opération d'insertion associée. De la même manière qu'une opération d'insertion, elle contient un groupe de messages MQGACF\_MESSAGE et, à l'intérieur de celui-ci, un groupe MQMD MQGACF\_MQMD. Toutefois, ce groupe MQMD contient uniquement les zones suivantes, qui peuvent être remplacées par un abonné: *Format, Priority, Persistence, MsgId, CorrelId, UserIdentifier, AccountingToken, ApplIdentityData*.

Les attributs *SubId* et *SubLevel* de l'abonné sont inclus dans les informations de l'opération. Vous pouvez utiliser *SubID* avec la commande MQCMD\_INQUIRE\_SUBSCRIBER PCF pour extraire tous les autres attributs d'un abonné.

L'opération de publication supprimée MQOPER\_DISCARDED\_PUBLISH est analogue à l'opération de suppression utilisée lorsqu'un message n'est pas distribué dans la messagerie point-à-point. Un message n'est pas distribué à un abonné si le message a été explicitement demandé de ne pas être distribué à une destination locale et que cet abonné spécifie une destination locale. Un message est également considéré comme non distribué en cas de problème lors de l'obtention du message dans la file d'attente de destination, par exemple parce que la file d'attente est saturée.

Les informations d'une opération de publication / suppression sont les mêmes que pour une opération de publication, avec l'ajout d'une zone *Commentaires* qui indique les raisons pour lesquelles le message n'a pas été distribué. Cette zone de retour contient les valeurs MQFB\_\* ou MQRC\_\* communes à l'opération MQOPER\_DISCARD. La raison de la suppression d'une publication, par opposition à son exclusion, est la même que celle de la suppression d'un put.

L'opération de publication exclue MQOPER\_EXCLUDED\_PUBLISH fournit des informations sur un abonné qui a été pris en compte pour la distribution du message, car la rubrique sur laquelle l'abonné s'abonne correspond à celle de l'opération d'insertion associée, mais le message n'a pas été distribué à l'abonné car d'autres critères de sélection ne correspondent pas au message qui est inséré dans la rubrique. Comme pour une opération de suppression de publication, la zone *Commentaires* fournit des informations sur la raison pour laquelle cet abonnement a été exclu. Toutefois, contrairement à l'opération de publication de suppression, aucune information relative aux messages n'est fournie car aucun message n'a été généré pour cet abonné.

### *Rapport de réponse d'insertion/insertion (MQOPER\_PUT/MQOPER\_PUT\_REPLY/MQOPER\_PUT\_REPORT)*

Les paramètres de données de message de rapport d'activité supplémentaires qui sont renvoyés dans le groupe PCF *Opération* pour le type d'opération Rapport d'insertion/insertion (MQOPER\_PUT/MQOPER\_PUT\_REPLY/MQOPER\_PUT\_REPORT) (un message, un message de réponse ou un message de rapport a été inséré dans une file d'attente).

#### **QName**

Description :	Nom de la file d'attente qui a été ouverte.
Identificateur :	MQCA_Q_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	LONGUEUR_NOM_Q_MQ_
Renvoyé:	Toujours, à l'exception d'une exception: non renvoyé si l'opération d'insertion concerne une rubrique, contenue dans une activité de publication.

#### **ResolvedQName**

Description :	Nom de la file d'attente ouverte.
Identificateur :	MQCACF_RESOLVED_Q_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	LONGUEUR_NOM_Q_MQ_
Renvoyé:	Lorsque la file d'attente ouverte a pu être résolue. Non renvoyé si l'opération d'insertion concerne une rubrique contenue dans une activité de publication.

#### **RemoteQName**

Description :	Nom de la file d'attente ouverte, tel qu'il est connu sur le gestionnaire de files d'attente éloignées.
Identificateur :	MQCA_REMOTE_Q_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	LONGUEUR_NOM_Q_MQ_
Renvoyé:	Si la file d'attente ouverte est une file d'attente éloignée. Non renvoyé si l'opération d'insertion concerne une rubrique contenue dans une activité de publication.

#### **RemoteQMgrName**

Description :	Nom du gestionnaire de files d'attente éloignées sur lequel la file d'attente éloignée est définie.
Identificateur :	MQCA_REMOTE_Q_MGR_NAME.
Type de données :	MQCFST.



Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH
Renvoyé:	Si la file d'attente ouverte est une file d'attente éloignée. Non renvoyé si l'opération d'insertion concerne une rubrique contenue dans une activité de publication.

### **TopicString**

Description :	Chaîne de rubrique complète dans laquelle le message est inséré.
Identificateur :	MQCA_TOPIC_CHAINE.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Si l'opération d'insertion concerne une rubrique, contenue dans une activité de publication.

### **Feedback**

Description :	Raison pour laquelle le message est inséré dans la file d'attente des messages non livrés.
Identificateur :	MQIACF_FEEDBACK.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Opération.</i>
Renvoyé:	Si le message a été inséré dans la file d'attente des messages non livrés.

### *Réception (MQOPER\_RECEIVE)*

Paramètres de données de message de rapport d'activité supplémentaires renvoyés dans le groupe PCF *Opération* pour le type d'opération Réception (MQOPER\_RECEIVE) (un message a été reçu sur un canal).

### **ChannelName**

Description :	Nom du canal sur lequel le message a été reçu.
Identificateur :	MQCACH_CHANNEL_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	LONGUEUR_NOM_CANAL_MQ_CANAL
Renvoyé:	Toujours.

### **ChannelType**

Description :	Type de canal sur lequel le message a été reçu.
Identificateur :	MQIACH_CHANNEL_TYPE.
Type de données :	MQCFIN.
Inclus dans le groupe PCF:	<i>Opération.</i>

Renvoyé: Toujours.

### **RemoteQMgrName**

Description : Nom du gestionnaire de files d'attente à partir duquel le message a été reçu.

Identificateur : MQCA\_REMOTE\_Q\_MGR\_NAME.

Type de données : MQCFST.

Inclus dans le groupe PCF: *Opération.*

Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

Renvoyé: Toujours.

### *Envoi (MQOPER\_SEND)*

Paramètres de données de message de rapport d'activité supplémentaires renvoyés dans le groupe PCF *Opération* pour le type d'opération Envoi (MQOPER\_SEND) (un message a été envoyé sur un canal).

### **ChannelName**

Description : Nom du canal où le message a été envoyé.

Identificateur : MQCACH\_CHANNEL\_NAME.

Type de données : MQCFST.

Inclus dans le groupe PCF: *Opération.*

Longueur maximale : MQ\_CHANNEL\_NAME\_LENGTH.

Renvoyé: Toujours.

### **ChannelType**

Description : Type de canal où le message a été envoyé.

Identificateur : MQIACH\_CHANNEL\_TYPE.

Type de données : MQCFIN.

Inclus dans le groupe PCF: *Opération.*

Renvoyé: Toujours.

### **XmitQName**

Description : File d'attente de transmission à partir de laquelle le message a été extrait.

Identificateur : MQCACH\_XMIT\_Q\_NAME.

Type de données : MQCFST.

Inclus dans le groupe PCF: *Opération.*

Longueur maximale : MQ\_Q\_NAME\_LENGTH.

Renvoyé: Toujours.

## **RemoteQMgrName**

Description :	Nom du gestionnaire de files d'attente éloignées auquel le message a été envoyé.
Identificateur :	MQCA_REMOTE_Q_MGR_NAME.
Type de données :	MQCFST.
Inclus dans le groupe PCF:	<i>Opération.</i>
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH
Renvoyé:	Toujours.

## **Référence du message de trace-route**

Cette page permet d'obtenir une vue d'ensemble du format des messages de trace. Les données de message de trace-route incluent des paramètres qui décrivent les activités causées par le message de trace-route

### **Format des messages de trace-route**

Les messages de trace-route sont des messages IBM MQ standard contenant un descripteur de message et des données de message. Les données de message contiennent des informations sur les activités effectuées sur un message d'acheminement de trace lorsqu'il a été acheminé via un réseau de gestionnaires de files d'attente.

Les messages de trace-route contiennent les informations suivantes:

#### **Un descripteur de message**

Une structure MQMD, avec la zone *Format* définie sur MQFMT\_ADMIN ou MQFMT\_EMBEDDED\_PCF.

#### **Données de message**

Se compose de l'un des éléments suivants:

- Un en-tête PCF (MQCFH) et des données de message de trace, si *Format* est défini sur MQFMT\_ADMIN, ou
- Un en-tête PCF imbriqué (MQEPH), des données de message de trace et des données de message supplémentaires spécifiées par l'utilisateur, si *Format* est défini sur MQFMT\_EMBEDDED\_PCF.

Lors de l'utilisation de l'application d'affichage de route IBM MQ pour générer un message de trace, *Format* est défini sur MQFMT\_ADMIN.

Le contenu des données de message de trace est déterminé par le paramètre *Accumuler* du groupe PCF *TraceRoute*, comme suit:

- Si *Cumuler* est défini sur MQROUTE\_ACCUMULATE\_NONE, les données de message de trace-route contiennent le groupe PCF *TraceRoute*.
- Si *Cumuler* est défini sur MQROUTE\_ACCUMULATE\_IN\_MSG ou MQROUTE\_ACCUMULATE\_AND\_REPLY, les données de message de trace contiennent le groupe PCF *TraceRoute* et zéro ou plusieurs groupes PCF *Activity*.

La [Tableau 20](#), à la page 132 montre la structure d'un message de suivi.

Tableau 20. Format des messages de trace-route

Structure MQMD	Structure MQEPH d'en-tête PCF intégrée	Données de message de trace-route
Identificateur de structure Version de la structure Options de rapport Type de message Délai d'expiration Commentaires Codage CCSID Format de message Priorité Persistance Identificateur de message Identificateur de corrélation Nombre d'annulations File de réponses Gestionnaire de file de réponses ID utilisateur Jeton de comptabilité Données sur l'identité de l'application Type d'application Nom d'application Date d'insertion Heure d'insertion Données sur l'origine de l'application Identificateur de groupe Numéro de séquence du message Décalage Indicateurs de message Longueur d'origine	Identificateur de structure Version de la structure Longueur de la structure Codage CCSID Format de message Indicateurs En-tête PCF (MQCFH) Type de structure Longueur de la structure Version de la structure Identificateur de commande Numéro de séquence du message Options de contrôle Code de fin d'exécution Code raison Nombre de paramètres	TraceRoute Détails Activités enregistrées Activités non enregistrées Nombre de discontinuités Activités max Cumuler Délivrer

### **Message de trace MQMD (descripteur de message)**

Cette page permet d'afficher les valeurs contenues dans la structure MQMD d'un message de suivi

#### **StrucId**

Description : Identificateur de structure.  
 Type de données : MQCHAR4.  
 Valeur : MQMD\_STRUC\_ID.

#### **Version**

Description : Numéro de version de la structure.  
 Type de données : MQLONG:  
 Valeurs : **MQMD\_VERSION\_1.**

#### **Report**

Description : Options des messages de rapport.

Type de données : MQLONG:  
Valeur : Défini en fonction des exigences. Les options de rapport communes sont les suivantes:

**MQRO\_DISCARD\_MSG**

Le message est supprimé à l'arrivée dans une file d'attente locale.

**MQRO\_PASS\_DISCARD\_ET\_EXPIRATION**

Chaque réponse (rapports d'activité ou message de réponse de suivi) aura l'option de rapport MQRO\_DISCARD\_MSG définie, et le délai d'expiration restant sera transmis. Cela garantit que les réponses ne restent pas indéfiniment dans le réseau du gestionnaire de files d'attente.

**MsgType**

Description : Type de message.  
Type de données : MQLONG:  
Valeur : Si le paramètre *Accumulation* du groupe TraceRoute est spécifié en tant que MQROUTE\_ACCUMULATE\_AND\_REPLY, le type de message est MQMT\_REQUEST  
Sinon :  
**MQMT\_DATAGRAM:**

**Expiry**

Description : Durée de vie du message.  
Type de données : MQLONG:  
Valeur : Défini en fonction des exigences. Ce paramètre permet de s'assurer que les messages de trace ne sont pas laissés indéfiniment dans un réseau de gestionnaires de files d'attente.

**Feedback**

Description : Retour d'informations ou code anomalie.  
Type de données : MQLONG:  
Valeur : **MQFB\_NONE:**

**Encoding**

Description : Codage numérique des données de message.  
Type de données : MQLONG:  
Valeur : Définisseau besoin.

**CodedCharSetId**

Description : Identificateur de jeu de caractères des données d'un message.  
Type de données : MQLONG:  
Valeur : Définisseau besoin.

**Format**

Description : Nom de format des données de message

Type de données : MQCHAR8.

Valeur :

**MQFMT\_ADMIN**

Message d'administration. Aucune donnée utilisateur ne suit le groupe PCF *TraceRoute* .

**MQFMT\_EMBEDDED\_PCF**

Message PCF imbriqué. Les données utilisateur suivent le groupe PCF *TraceRoute* .

**Priority**

Description : Priorité du message.

Type de données : MQLONG:

Valeur : Défini en fonction des exigences.

**Persistence**

Description : Persistance des messages.

Type de données : MQLONG:

Valeur : Défini en fonction des exigences.

**MsgId**

Description : Identificateur de message.

Type de données : MQBYTE24.

Valeur : Défini en fonction des exigences.

**CorrelId**

Description : Identificateur de corrélation.

Type de données : MQBYTE24.

Valeur : Défini en fonction des exigences.

**BackoutCount**

Description : Compteur d'annulation.

Type de données : MQLONG:

Valeur : 0.

**ReplyToQ**

Description : Nom de la file d'attente de réponses.

Type de données : MQCHAR48.

Valeurs : Défini en fonction des exigences.

Si *MsgType* est défini sur MQMT\_REQUEST ou si *Report* a des options de génération de rapport définies, ce paramètre doit être non vide.

**ReplyToQMgr**

Description : Nom du gestionnaire de files d'attente de réponses.

Type de données : MQCHAR48.

Valeur : Défini en fonction des exigences.

### ***UserIdentifier***

Description : Identificateur utilisateur de l'application à l'origine du message.

Type de données : MQCHAR12.

Valeur : Défini comme normal.

### ***AccountingToken***

Description : Jeton de comptabilité qui permet à une application de facturer le travail effectué à la suite du message.

Type de données : MQBYTE32.

Valeur : Défini comme normal.

### ***ApplIdentityData***

Description : Données d'application relatives à l'identité.

Type de données : MQCHAR32.

Valeurs : Défini comme normal.

### ***PutApplType***

Description : Type d'application qui a inséré le message.

Type de données : MQLONG.

Valeur : Défini comme normal.

### ***PutApplName***

Description : Nom de l'application qui a inséré le message.

Type de données : MQCHAR28.

Valeur : Défini comme normal.

### ***PutDate***

Description : Date à laquelle le message a été inséré.

Type de données : MQCHAR8.

Valeur : Défini comme normal.

### ***PutTime***

Description : Heure à laquelle le message a été inséré.

Type de données : MQCHAR8.

Valeur : Défini comme normal.

### ***ApplOriginData***

Description : Données d'application relatives à l'origine.

Type de données : MQCHAR4.

Valeur : Définir comme normal ..

### **Message de trace MQEPH (en-tête PCF imbriqué)**

Cette page permet d'afficher les valeurs contenues dans la structure MQEPH pour un message de suivi

La structure MQEPH contient une description des informations PCF qui accompagnent les données de message d'un message de trace et des données de message d'application qui le suivent. Une structure MQEPH est utilisée uniquement si des données de message utilisateur supplémentaires suivent le groupe PCF TraceRoute .

Pour un message de suivi, la structure MQEPH contient les valeurs suivantes:

#### **StrucId**

Description :        Identificateur de structure.  
Type de données :    MQCHAR4.  
Valeur :             MQEPH\_STRUC\_ID.

#### **Version**

Description :        Numéro de version de la structure.  
Type de données :    MQLONG:  
Valeurs :            MQEPH\_VERSION\_1.

#### **StrucLength**

Description :        Longueur de la structure.  
Type de données :    MQLONG:  
Valeur :             Longueur totale de la structure, y compris les structures de paramètre PCF qui la suivent.

#### **Encoding**

Description :        Codage numérique des données de message qui suivent la dernière structure de paramètre PCF.  
Type de données :    MQLONG:  
Valeur :             Codage des données de message.

#### **CodedCharSetId**

Description :        Identificateur de jeu de caractères des données de message qui suivent la dernière structure de paramètre PCF.  
Type de données :    MQLONG:  
Valeur :             Jeu de caractères des données de message.

#### **Format**

Description :        Nom de format des données de message qui suivent la dernière structure de paramètre PCF.  
Type de données :    MQCHAR8.  
Valeur :             Nom de format des données de message.

#### **Flags**

Description :        Indicateurs spécifiant les attributs de la structure ou contrôlant son traitement.  
Type de données :    MQLONG:



Valeur : **MQEPH\_NONE**  
Aucun indicateur spécifié.

**MQEPH\_CCSID\_IMBRIQUÉ**  
Indique que le jeu de caractères des paramètres contenant des données de type caractères est spécifié individuellement dans la zone *CodedCharSetId* de chaque structure.

### **PCFHeader**

Description : En-tête de format de commande programmable

Type de données : MQCFH.

Valeur : Voir «Message de trace MQCFH (en-tête PCF)», à la page 137.

### **Message de trace MQCFH (en-tête PCF)**

Cette page permet d'afficher les valeurs PCF contenues dans la structure MQCFH pour un message de trace-route

Pour un message de trace-route, la structure MQCFH contient les valeurs suivantes:

#### **Type**

Description : Type de structure qui identifie le contenu du message.

Type de données : MQLONG:

Valeur : **MQCFT\_TRACE\_ROUTE**  
Le message est un message de suivi.

#### **StrucLength**

Description : Longueur de la structure.

Type de données : MQLONG:

Valeur : **LONGUEUR\_STRUC\_MQCFH**  
Longueur en octets de la structure MQCFH.

#### **Version**

Description : Numéro de version de la structure.

Type de données : MQLONG:

Valeurs : MQCFH\_VERSION\_3

#### **Command**

Description : Identificateur de commande. Identifie la catégorie du message.

Type de données : MQLONG:

Valeurs : **MQCMD\_TRACE\_ROUTE**  
Message de trace-route.

#### **MsgSeqNumber**

Description : Numéro de séquence du message. Il s'agit du numéro de séquence du message dans un groupe de messages associés.

Type de données : MQLONG:

Valeurs : 1.

### **Control**

Description : Options de contrôle.  
Type de données : MQLONG:  
Valeurs : MQCFC\_LAST.

### **CompCode**

Description : Code achèvement.  
Type de données : MQLONG:  
Valeurs : MQCC\_OK.

### **Reason**

Description : Code raison qualifiant le code achèvement.  
Type de données : MQLONG:  
Valeurs : MQRC\_NONE:

### **ParameterCount**

Description : Nombre de structures de paramètres. Il s'agit du nombre de structures de paramètres qui suivent la structure MQCFH. Une structure de groupe (MQCFGR) et ses structures de paramètres incluses sont comptées comme une seule structure.  
Type de données : MQLONG:  
Valeurs : 1 ou supérieur.

### **Données de message de trace-route**

Cette page permet d'afficher les paramètres qui constituent la partie du groupe PCF *TraceRoute* des données de message de trace

Le contenu des données de message de trace dépend du paramètre *Accumuler* du groupe PCF *TraceRoute* . Les données de message de trace sont constituées du groupe PCF *TraceRoute* et de zéro ou plusieurs groupes PCF *Activity* . Le groupe PCF *TraceRoute* est détaillé dans cette rubrique. Pour plus d'informations sur le groupe PCF *Activity* , voir les informations connexes.

Les données de message de trace-route contiennent les paramètres suivants:

### **TraceRoute**

Description : Paramètres groupés spécifiant les attributs du message de trace-route. Pour un message de trace-route, certains de ces paramètres peuvent être modifiés pour contrôler la façon dont il est traité.  
Identificateur : MQGACF\_TRACE\_ROUTE.  
Type de données : MQCFGR.  
Contenu dans le groupe PCF: Néant.

Paramètres du groupe: *Detail*  
*RecordedActivities*  
*UnrecordedActivities*  
*DiscontinuityCount*  
*MaxActivities*  
*Accumulate*  
*Forward*  
*Deliver*

### ***Detail***

Description : Niveau de détail qui sera enregistré pour l'activité.

Identificateur : MQIACF\_ROUTE\_DETAIL.

Type de données : MQCFIN.

Contenu dans le groupe PCF: *TraceRoute*.

Valeurs : **MQROUTE\_DETAIL\_LOW**  
Les activités effectuées par l'application écrite par l'utilisateur sont enregistrées.

**MQROUTE\_DETAIL\_MEDIUM**  
Les activités spécifiées dans MQROUTE\_DETAIL\_LOW sont enregistrées. A celles-ci s'ajoutent les activités effectuées par les agents MCA.

**MQROUTE\_DETAIL\_ELEVE**  
Les activités spécifiées dans MQROUTE\_DETAIL\_LOW et MQROUTE\_DETAIL\_MEDIUM sont enregistrées. Les MCM n'enregistrent aucune autre information sur l'activité à ce niveau de détail. Cette option est disponible uniquement pour les applications écrites par l'utilisateur qui doivent enregistrer des informations supplémentaires sur l'activité.

### ***RecordedActivities***

Description : Nombre d'activités à l'origine du message de trace-route, où les informations ont été enregistrées.

Identificateur : MQIACF\_RECORDED\_ACTIVITÉS.

Type de données : MQCFIN.

Contenu dans le groupe PCF: *TraceRoute*.

### ***UnrecordedActivities***

Description : Nombre d'activités causées par le message de trace-route, pour lesquelles les informations n'ont pas été enregistrées.

Identificateur : MQIACF\_UNRECORDED\_ACTIVITÉS.

Type de données : MQCFIN.

Contenu dans le groupe PCF: *TraceRoute*.

### ***DiscontinuityCount***

Description : Nombre de fois où un message de trace-route a été reçu d'un gestionnaire de files d'attente qui ne prend pas en charge la messagerie de trace-route.

Identificateur : MQIACF\_DISCONTINUITY\_COUNT.  
Type de données : MQCFIN.  
Contenu dans le groupe PCF: *TraceRoute.*

### **MaxActivities**

Description : Nombre maximal d'activités dans lesquelles le message de suivi peut être impliqué avant qu'il ne cesse d'être traité.

Identificateur : MQIACF\_MAX\_ACTIVITÉS.

Type de données : MQCFIN.

Contenu dans le groupe PCF: *TraceRoute.*

Valeur : **Entier positif**  
Nombre maximal d'activités.

**MQROUTE\_UNLIMITED\_ACTIVITÉS**  
Un nombre illimité d'activités.

### **Accumulate**

Description : Indique si les informations d'activité sont cumulées dans le message de routage de trace et si un message de réponse contenant les informations d'activité cumulées est généré avant que le message de routage de trace soit supprimé ou placé dans une file d'attente de non transmission.

Identificateur : MQIACF\_ROUTE\_ACCUMULATION.

Type de données : MQCFIN.

Contenu dans le groupe PCF: *TraceRoute.*

Valeur : **MQROUTE\_ACCUMULATE\_NONE**  
Les informations d'activité ne sont pas accumulées dans les données de message du message de suivi.

**MQROUTE\_ACCUMULATE\_EN\_MSG**  
Les informations d'activité sont accumulées dans les données de message du message de suivi.

**MQROUTE\_ACCUMULATE\_ET\_REPONSE**  
Les informations d'activité sont accumulées dans les données de message du message de suivi et un message de réponse de suivi est généré.

### **Forward**

Description : Indique les gestionnaires de files d'attente vers lesquels le message de suivi peut être réacheminé. Lorsqu'ils déterminent si un message doit être réacheminé vers un gestionnaire de files d'attente éloignées, les gestionnaires de files d'attente utilisent l'algorithme décrit dans la rubrique [Transfert](#).

Identificateur : MQIACF\_ROUTE\_FORWARDING.

Type de données : MQCFIN.

Contenu dans le groupe PCF: *TraceRoute.*

Valeur :	<b>MQROUTE_WARD_IF_SUPPORTED</b> Le message de trace-route est réacheminé uniquement vers les gestionnaires de files d'attente qui respecteront la valeur du paramètre <i>Distribution</i> du groupe <i>TraceRoute</i> .
	<b>MQROUTE_ACHEMINEMENT_TOUT</b> Le message de trace-route est réacheminé vers n'importe quel gestionnaire de files d'attente, que la valeur du paramètre <i>Distribution</i> soit ou non respectée.

### ***Deliver***

Description :	Indique l'action à effectuer si le message de suivi arrive dans la file d'attente de destination.
Identificateur :	MQIACF_ROUTE_DELIVERY.
Type de données :	MQCFIN.
Contenu dans le groupe PCF:	<i>TraceRoute</i> .
Valeur :	<b>MQROUTE_DELIVER_YES</b> A l'arrivée, le message de suivi est placé dans la file d'attente cible. Toute application exécutant une opération d'extraction destructive sur la file d'attente cible peut recevoir le message de trace-route.
	<b>MQROUTE_NO</b> A l'arrivée, le message de suivi est supprimé.

## **Référence du message de réponse de trace-route**

Cette page permet d'obtenir une présentation du format de message de réponse de trace-route. Les données de message de réponse de trace-route sont un doublon des données de message de trace-route provenant du message de trace-route pour lequel elles ont été générées.

### ***Format du message de réponse de trace-route***

Les messages de réponse de trace-route sont des messages IBM MQ standard contenant un descripteur de message et des données de message. Les données de message contiennent des informations sur les activités effectuées sur un message d'acheminement de trace lorsqu'il a été acheminé via un réseau de gestionnaires de files d'attente.

Les messages de réponse de routage de trace contiennent les informations suivantes:

#### **Un descripteur de message**

Une structure MQMD

#### **Données de message**

Un en-tête PCF (MQCFH) et des données de message de réponse de trace

Les données de message de réponse d'acheminement de trace se composent d'un ou de plusieurs groupes PCF *Activity* .

Lorsqu'un message de trace-route atteint sa file d'attente cible, un message de réponse de trace-route peut être généré qui contient une copie des informations d'activité du message de trace-route. Le message de réponse de trace-route sera distribué à une file d'attente de réponse ou à une file d'attente système.

Le [Tableau 21](#), à la page 142 présente la structure d'un message de réponse de suivi, y compris les paramètres qui ne sont renvoyés que dans certaines conditions.

Tableau 21. Format du message de réponse de trace-route

Structure MQMD	Structure MQCFH d'en-tête PCF	Données de message de réponse de trace-route
Identificateur de structure Version de la structure Options de rapport Type de message Délai d'expiration Commentaires Codage CCSID Format de message Priorité Persistance Identificateur de message Identificateur de corrélation Nombre d'annulations File de réponses Gestionnaire de file de réponses ID utilisateur Jeton de comptabilité Données sur l'identité de l'application Type d'application Nom d'application Date d'insertion Heure d'insertion Données sur l'origine de l'application Identificateur de groupe Numéro de séquence du message Décalage Indicateurs de message Longueur d'origine	En-tête PCF (MQCFH) Type de structure Longueur de la structure Version de la structure Identificateur de commande Numéro de séquence du message Options de contrôle Code de fin d'exécution Code raison Nombre de paramètres	Activité Nom de l'application d'activité Type d'application d'activité Description de l'activité Opération Type d'opération Date de l'opération Heure de l'opération Message LONGUEUR DE MESSAGE MQMD EmbeddedMQMD Nom gest. de files Nom du groupe de partage de files d'attente Nom de la file d'attente <sup>1 2 3</sup> Nom de file d'attente résolu <sup>1 3</sup> Nom de la file d'attente éloignée <sup>3</sup> Gestionnaire de files d'attente éloignées-nom <sup>2 3 4 5</sup> Commentaires en retour <sup>2</sup> Nom de canal <sup>4 5</sup> Type de canal <sup>4 5</sup> Nom de la file d'attente de transmission <sup>5</sup> TraceRoute Détails Activités enregistrées Activités non enregistrées Nombre de discontinuités Activités max Cumuler Délivrer

**Remarque :**

1. Renvoyé pour les opérations d'extraction et de navigation.
2. Renvoyé pour les opérations de suppression.
3. Renvoyé pour les opérations d'insertion, de réponse d'insertion et de rapport d'insertion.
4. Renvoyé pour les opérations de réception.
5. Renvoyé pour les opérations d'envoi.

**Message de réponse de suivi MQMD (descripteur de message)**

Cette page permet d'afficher les valeurs contenues dans la structure MQMD pour un message de réponse de suivi

Pour un message de réponse de type trace-route, la structure MQMD contient les paramètres décrits dans Descripteur de message de rapport d'activité. Certaines valeurs de paramètre d'un descripteur de

message de réponse de trace-route sont différentes de celles d'un descripteur de message de rapport d'activité, comme suit:

### **MsgType**

Description : Type de message.  
Type de données : MQLONG:  
Valeur : **MQMT\_REPLY**

### **Feedback**

Description : Retour d'informations ou code anomalie.  
Type de données : MQLONG:  
Valeur : **MQFB\_AUCUN**

### **Encoding**

Description : Codage numérique des données de message.  
Type de données : MQLONG:  
Valeur : Copié à partir du descripteur de message de trace-route.

### **CodedCharSetId**

Description : Identificateur de jeu de caractères des données d'un message.  
Type de données : MQLONG:  
Valeur : Copié à partir du descripteur de message de trace-route.

### **Format**

Description : Nom de format des données de message  
Type de données : MQCHAR8.  
Valeur : **MQFMT\_ADMIN**  
Message d'administration.

### **Message de réponse de suivi MQCFH (en-tête PCF)**

Cette page permet d'afficher les valeurs PCF contenues dans la structure MQCFH pour un message de réponse de type trace-route

L'en-tête PCF (MQCFH) d'un message de réponse de type trace-route est identique à celui d'un message de type trace-route.

### **Données de message de réponse de trace-route**

Les données de message de réponse de trace-route sont un doublon des données de message de trace-route provenant du message de trace-route pour lequel elles ont été générées.

Les données de message de réponse de trace-route contiennent un ou plusieurs groupes *Activity* . Les paramètres sont décrits dans le [«Données de message du rapport d'activité»](#), à la page 113.

## **Messages de comptabilité et de statistiques**


Les gestionnaires de files d'attente génèrent des messages de comptabilité et de statistiques pour enregistrer des informations sur les opérations MQI effectuées par les applications IBM MQ ou pour enregistrer des informations sur les activités se produisant dans un système IBM MQ .

## messages de comptabilité

Les messages de comptabilité sont utilisés pour enregistrer des informations sur les opérations MQI effectuées par les applications IBM MQ . Voir [«messages de comptabilité»](#), à la page 144.

## messages sur les statistiques

Les messages de statistiques sont utilisés pour enregistrer des informations sur les activités qui se produisent dans un système IBM MQ . Voir [«messages sur les statistiques»](#), à la page 148. Certaines activités enregistrées dans les messages de statistiques sont liées à des opérations internes du gestionnaire de files d'attente.

 Les messages de comptabilité et les messages de statistiques décrits ici ne sont pas disponibles sous IBM MQ for z/OS, mais des fonctionnalités équivalentes sont disponibles via la fonction SMF (System Management Facility).

Les messages de comptabilité et de statistiques sont distribués à l'une des deux files d'attente système. Les applications utilisateur peuvent extraire les messages de ces files d'attente système et utiliser les informations enregistrées à diverses fins:

- Compter pour l'utilisation des ressources d'application.
- Enregistrer l'activité de l'application.
- Planification des capacités
- Détectez les problèmes dans votre réseau de gestionnaires de files d'attente.
- Aide à déterminer les causes des problèmes dans votre réseau de gestionnaires de files d'attente.
- Améliorez l'efficacité de votre réseau de gestionnaires de files d'attente.
- Familiarisez-vous avec l'exécution de votre réseau de gestionnaires de files d'attente.
- Vérifiez que votre réseau de gestionnaires de files d'attente s'exécute correctement.

## Concepts associés

[«Using System Management Facility»](#), à la page 335

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

## messages de comptabilité

Les messages de comptabilité enregistrent des informations sur les opérations MQI effectuées par les applications IBM MQ . Un message de comptabilité est un message PCF qui contient un certain nombre de structures PCF.

Lorsqu'une application se déconnecte d'un gestionnaire de files d'attente, un message de comptabilité est généré et distribué à la file d'attente de comptabilité système (SYSTEM.ADMIN.ACCOUNTING.QUEUE). Pour les applications IBM MQ à exécution longue, les messages de comptabilité intermédiaires sont générés comme suit:

- Lorsque le temps écoulé depuis l'établissement de la connexion dépasse l'intervalle configuré.
- Lorsque le temps écoulé depuis le dernier message de comptabilité intermédiaire dépasse l'intervalle configuré.

Les messages de comptabilité se trouvent dans les catégories suivantes:

### Messages de comptabilité MQI

Les messages de comptabilité MQI contiennent des informations relatives au nombre d'appels MQI effectués à l'aide d'une connexion à un gestionnaire de files d'attente.

### Messages de comptabilité de file d'attente

Les messages de comptabilité de file d'attente contiennent des informations relatives au nombre d'appels MQI effectués à l'aide de connexions à un gestionnaire de files d'attente, regroupées par file d'attente.



Chaque message de comptabilité de file d'attente peut contenir jusqu'à 100 enregistrements, chaque enregistrement relatif à une activité effectuée par l'application par rapport à une file d'attente spécifique.

Les messages de comptabilité sont enregistrés uniquement pour les files d'attente locales. Si une application effectue un appel MQI sur une file d'attente alias, les données de comptabilité sont enregistrées sur la file d'attente de base et, pour une file d'attente éloignée, les données de comptabilité sont enregistrées sur la file d'attente de transmission.

**Remarque :** Etant donné que les informations concernent les opérations MQI effectuées par les applications IBM MQ, elles n'incluent pas les opérations relatives à une file d'attente de diffusion en flux, sauf si l'opération se trouve directement sur cette file d'attente par l'application.

### Référence associée

«Données de message de comptabilité MQI», à la page 165

Cette page permet d'afficher la structure d'un message de comptabilité MQI

«Données de message de comptabilité de file d'attente», à la page 177

Cette page permet d'afficher la structure d'un message de comptabilité de file d'attente

### Format des messages de comptabilité

Les messages de comptabilité comprennent un ensemble de champs PCF constitués d'un descripteur de message et de données de message.

#### Descripteur de message

- Un message de comptabilité MQMD (descripteur de message)

#### Données de message de comptabilité

- Un message de comptabilité MQCFH (en-tête PCF)
- Données de message de comptabilité toujours renvoyées
- Données de message de comptabilité renvoyées si elles sont disponibles

Le message de comptabilité MQCFH (en-tête PCF) contient des informations sur l'application et l'intervalle pour lequel les données de comptabilité ont été enregistrées.

Les données de message de comptabilité comprennent les paramètres PCF qui stockent les informations de comptabilité. Le contenu des messages de comptabilité dépend de la catégorie de message comme suit:

#### Message de comptabilité MQI

Les données de message de comptabilité MQI se composent d'un certain nombre de paramètres PCF, mais pas de groupes PCF.

#### Message de comptabilité de file d'attente

Les données de message de comptabilité de file d'attente se composent d'un certain nombre de paramètres PCF et sont comprises entre 1 et 100 groupes PCF *QAccountingData*.

Il existe un groupe PCF *QAccountingData* pour chaque file d'attente pour laquelle des données comptables ont été collectées. Si une application accède à plus de 100 files d'attente, plusieurs messages de comptabilité sont générés. Le *SeqNumber* de chaque message dans l'en-tête MQCFH (PCF) est mis à jour en conséquence et le dernier message de la séquence comporte le paramètre *Contrôle* dans le MQCFH spécifié sous la forme MQCFC\_LAST.

### Collecte des informations comptables

Utilisez les attributs de file d'attente et de gestionnaire de files d'attente pour contrôler la collecte des informations de comptabilité. Vous pouvez également utiliser les options MQCONNX pour contrôler la collecte au niveau de la connexion.

#### Contrôle de la collecte des informations de comptabilité MQI

Utilisez l'attribut de gestionnaire de files d'attente ACCTMQI pour contrôler la collecte des informations de comptabilité MQI.

Pour modifier la valeur de cet attribut, utilisez la commande MQSC, ALTER QMGR, et spécifiez le paramètre ACCTMQI. Les messages de comptabilité sont générés uniquement pour les connexions qui commencent après l'activation de la comptabilité. Le paramètre **ACCTMQI** peut avoir les valeurs suivantes:

**Oui**

Les informations de comptabilité MQI sont collectées pour chaque connexion au gestionnaire de files d'attente.

**Non**

Les informations de comptabilité MQI ne sont pas collectées. Il s'agit de la valeur par défaut.

Par exemple, pour activer la collecte des informations de comptabilité MQI, utilisez la commande MQSC suivante:

```
ALTER QMGR ACCTMQI(ON)
```

*Informations de comptabilité de file d'attente*

Utilisez l'attribut de file d'attente ACCTQ et l'attribut de gestionnaire de files d'attente ACCTQ pour contrôler la collecte des informations de comptabilité de file d'attente.

Pour modifier la valeur de l'attribut de file d'attente, utilisez la commande MQSC, ALTER QLOCAL, et spécifiez le paramètre ACCTQ. Les messages de comptabilité sont générés uniquement pour les connexions qui commencent après l'activation de la comptabilité. Notez que les modifications apportées à cette valeur ne sont effectives que pour les connexions au gestionnaire de files d'attente qui se produisent après la modification de l'attribut.

L'attribut de file d'attente ACCTQ peut avoir les valeurs suivantes:

**Oui**

Les informations de comptabilité de file d'attente pour cette file d'attente sont collectées pour chaque connexion au gestionnaire de files d'attente qui ouvre la file d'attente.

**Non**

Les informations de comptabilité de file d'attente pour cette file d'attente ne sont pas collectées.

**QMGR**

La collecte des informations de comptabilité de file d'attente pour cette file d'attente est contrôlée en fonction de la valeur de l'attribut de gestionnaire de files d'attente ACCTQ. Il s'agit de la valeur par défaut.

Pour modifier la valeur de l'attribut de gestionnaire de files d'attente, utilisez la commande MQSC ALTER QMGR et spécifiez le paramètre ACCTQ. L'attribut de gestionnaire de files d'attente ACCTQ peut avoir les valeurs suivantes:

**Oui**

Les informations de comptabilité des files d'attente sont collectées pour les files d'attente dont l'attribut de file d'attente ACCTQ est défini comme QMGR.

**Non**

Les informations de comptabilité des files d'attente ne sont pas collectées pour les files dont l'attribut de file d'attente ACCTQ est défini comme QMGR. Il s'agit de la valeur par défaut.

**Aucun**

La collecte des informations de comptabilité de file d'attente est désactivée pour toutes les files d'attente, quel que soit l'attribut de file d'attente ACCTQ.

Si l'attribut de gestionnaire de files d'attente, ACCTQ, est défini sur NONE, la collecte des informations de comptabilité de file d'attente est désactivée pour toutes les files d'attente, quel que soit l'attribut de file d'attente ACCTQ.

Par exemple, pour activer la collecte des informations de comptabilité pour la file d'attente, Q1, utilisez la commande MQSC suivante:

```
ALTER QLOCAL(Q1) ACCTQ(ON)
```

Pour activer la collecte des informations de comptabilité pour toutes les files d'attente qui spécifient l'attribut de file d'attente ACCTQ comme QMGR, utilisez la commande MQSC suivante:

```
ALTER QMGR ACCTQ(ON)
```

#### *Options MQCONNX pour le contrôle de la collecte des informations de comptabilité*

Utilisez le paramètre **ConnectOpts** dans l'appel MQCONNX pour modifier la collecte des informations de comptabilité de MQI et de file d'attente au niveau de la connexion en remplaçant les valeurs effectives des attributs de gestionnaire de files d'attente ACCTMQI et ACCTQ

Le paramètre **ConnectOpts** peut avoir les valeurs suivantes:

#### **MQCNO\_ACCOUNTING\_MQI\_ENABLED**

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTMQI est définie sur OFF, la comptabilité MQI est activée pour cette connexion. Il s'agit de l'équivalent de l'attribut de gestionnaire de files d'attente ACCTMQI défini sur ON.

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTMQI n'est pas définie sur OFF, cet attribut n'a aucun effet.

#### **MQCNO\_ACCOUNTING\_MQI\_DISABLED**

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTMQI est définie sur ON, la comptabilité MQI est désactivée pour cette connexion. Il s'agit de l'équivalent de l'attribut de gestionnaire de files d'attente ACCTMQI défini sur OFF.

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTMQI n'est pas définie sur ON, cet attribut n'a aucun effet.

#### **MQCNO\_ACCOUNTING\_Q\_ENABLED**

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTQ est définie sur OFF, la comptabilité de file d'attente est activée pour cette connexion. Toutes les files d'attente pour lesquelles ACCTQ est indiqué comme QMGR sont activées pour la comptabilité des files d'attente. Il s'agit de l'équivalent de l'attribut de gestionnaire de files d'attente ACCTQ défini sur ON.

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTQ n'est pas définie sur OFF, cet attribut n'a aucun effet.

#### **MQCNO\_ACCOUNTING\_Q\_DISABLED**

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTQ est définie sur ON, la comptabilité de file d'attente est désactivée pour cette connexion. Il s'agit de l'équivalent de l'attribut de gestionnaire de files d'attente ACCTQ défini sur OFF.

Si la valeur de l'attribut de gestionnaire de files d'attente ACCTQ n'est pas définie sur ON, cet attribut n'a aucun effet.

Ces substitutions sont désactivées par défaut. Pour les activer, définissez l'attribut de gestionnaire de files d'attente ACCTCONO sur ENABLED. Pour activer les substitutions de comptabilité pour les connexions individuelles, utilisez la commande MQSC suivante:

```
ALTER QMGR ACCTCONO(ENABLED)
```

#### *Génération de messages de comptabilité lorsque les applications se déconnectent*

Des messages de comptabilité sont générés lorsqu'une application se déconnecte du gestionnaire de files d'attente. Les messages de comptabilité intermédiaires sont également écrits pour les applications IBM MQ à exécution longue.

Les messages de comptabilité sont générés de l'une des manières suivantes lorsqu'une application se déconnecte:

- L'application émet un appel MQDISC
- Le gestionnaire de files d'attente reconnaît que l'application s'est arrêtée

Les messages de comptabilité intermédiaire sont écrits pour les applications IBM MQ à exécution longue lorsque l'intervalle depuis l'établissement de la connexion ou depuis le dernier message de comptabilité intermédiaire qui a été écrit dépasse l'intervalle configuré. L'attribut de gestionnaire de files d'attente, ACCTINT, indique la durée, en secondes, au bout de laquelle les messages de comptabilité intermédiaires peuvent être écrits automatiquement. Les messages de comptabilité sont générés uniquement lorsque l'application interagit avec le gestionnaire de files d'attente, de sorte que les applications qui restent connectées au gestionnaire de files d'attente pendant de longues périodes sans exécuter de demandes MQI ne génèrent pas de messages de comptabilité jusqu'à l'exécution de la première demande MQI à la fin de l'intervalle de comptabilité.

L'intervalle de comptabilité par défaut est de 1800 secondes (30 minutes). Par exemple, pour remplacer l'intervalle de comptabilité par 900 secondes (15 minutes), utilisez la commande MQSC suivante:

```
ALTER QMGR ACCTINT(900)
```

## messages sur les statistiques

Les messages de statistiques enregistrent des informations sur les activités qui se produisent dans un système IBM MQ . Un message de statistiques est un message PCF qui contient un certain nombre de structures PCF.

Les messages de statistiques sont distribués dans la file d'attente système (SYSTEM.ADMIN.STATISTICS.QUEUE) à des intervalles configurés, chaque fois qu'il y a une activité.

Les messages de statistiques se trouvent dans les catégories suivantes:

### messages sur les statistiques MQI

Les messages de statistiques MQI contiennent des informations relatives au nombre d'appels MQI effectués au cours d'un intervalle configuré. Par exemple, les informations peuvent inclure le nombre d'appels MQI émis par un gestionnaire de files d'attente.

### Messages de statistiques de file d'attente

Les messages de statistiques de file d'attente contiennent des informations relatives à l'activité d'une file d'attente pendant un intervalle configuré. Ces informations incluent le nombre de messages insérés dans la file d'attente et extraits de celle-ci, ainsi que le nombre total d'octets traités par une file d'attente.

Chaque message de statistiques de file d'attente peut contenir jusqu'à 100 enregistrements, chaque enregistrement étant lié à l'activité par file d'attente pour laquelle des statistiques ont été collectées.

Les messages de statistiques sont enregistrés uniquement pour les files d'attente locales. Si une application effectue un appel MQI sur une file d'attente alias, les données statistiques sont enregistrées sur la file d'attente de base et, pour une file d'attente éloignée, les données statistiques sont enregistrées sur la file d'attente de transmission.

### Messages de statistiques de canal

Les messages de statistiques de canal contiennent des informations relatives à l'activité d'un canal pendant un intervalle configuré. Par exemple, les informations peuvent être le nombre de messages transférés par le canal ou le nombre d'octets transférés par le canal.

Chaque message de statistiques de canal contient jusqu'à 100 enregistrements, chaque enregistrement étant lié à l'activité par canal pour lequel des statistiques ont été collectées.

**Remarque :** Etant donné que les informations concernent les activités se produisant dans un système IBM MQ , elles incluent les opérations relatives aux files d'attente de diffusion en flux.

## Référence associée

«Contrôle de la collecte de statistiques MQI», à la page 150

Utilisez l'attribut de gestionnaire de files d'attente STATMQI pour contrôler la collecte des informations de statistiques MQI.

«Contrôle de la collecte des statistiques de file d'», à la page 150

Utilisez l'attribut de file d'attente STATQ et l'attribut de gestionnaire de files d'attente STATQ pour contrôler la collecte des informations de statistiques de file d'attente

«Contrôle de la collecte des statistiques de canal», à la page 151

Utilisez l'attribut de canal STATCHL pour contrôler la collecte des informations de statistiques de canal. Vous pouvez également définir des attributs de gestionnaire de files d'attente pour contrôler la collecte d'informations.

## Format des messages de statistiques

Les messages statistiques comprennent un ensemble de champs PCF constitués d'un descripteur de message et de données de message.

### Descripteur de message

- Un message de statistiques MQMD (descripteur de message)

### Données de message de comptabilité

- Un message de statistiques MQCFH (en-tête PCF)
- Données de message de statistiques toujours renvoyées
- Données de message de statistiques renvoyées si elles sont disponibles

Le message de statistiques MQCFH (en-tête PCF) contient des informations sur l'intervalle pendant lequel les données statistiques ont été enregistrées.

Les données de message de statistiques comprennent les paramètres PCF qui stockent les informations de statistiques. Le contenu des messages de statistiques dépend de la catégorie de message comme suit:

### Message de statistiques MQI

Les données de message de statistiques MQI se composent d'un certain nombre de paramètres PCF, mais pas de groupes PCF.

### Message de statistiques de file d'attente

Les données de message de statistiques de file d'attente se composent d'un certain nombre de paramètres PCF et sont comprises entre 1 et 100 groupes PCF *QStatisticsData* .

Il existe un groupe PCF *QStatisticsData* pour chaque file d'attente active dans l'intervalle. Si plus de 100 files d'attente étaient actives dans l'intervalle, plusieurs messages de statistiques sont générés. Le *SeqNumber* de chaque message dans l'en-tête MQCFH (PCF) est mis à jour en conséquence et le dernier message de la séquence comporte le paramètre *Contrôle* dans le MQCFH spécifié sous la forme MQCFC\_LAST.

### Message de statistiques de canal

Les données de message de statistiques de canal se composent d'un certain nombre de paramètres PCF et sont comprises entre 1 et 100 groupes PCF *ChlStatisticsData* .

Il existe un groupe PCF *ChlStatisticsData* pour chaque canal qui était actif dans l'intervalle. Si plus de 100 canaux étaient actifs dans l'intervalle, plusieurs messages de statistiques sont générés. Le *SeqNumber* de chaque message dans l'en-tête MQCFH (PCF) est mis à jour en conséquence et le dernier message de la séquence comporte le paramètre *Contrôle* dans le MQCFH spécifié sous la forme MQCFC\_LAST.

## Collecte d'informations statistiques

Utilisez les attributs de file d'attente, de gestionnaire de files d'attente et de canal pour contrôler la collecte des informations statistiques

### *Contrôle de la collecte de statistiques MQI*

Utilisez l'attribut de gestionnaire de files d'attente STATMQI pour contrôler la collecte des informations de statistiques MQI.

Pour modifier la valeur de cet attribut, utilisez la commande MQSC ALTER QMGR et spécifiez le paramètre **STATMQI**. Les messages de statistiques sont générés uniquement pour les files d'attente ouvertes après l'activation de la collecte de statistiques. Le paramètre **STATMQI** peut avoir les valeurs suivantes:

#### **Oui**

Les informations de statistiques MQI sont collectées pour chaque connexion au gestionnaire de files d'attente.

#### **Non**

Les informations de statistiques MQI ne sont pas collectées. Il s'agit de la valeur par défaut.

Par exemple, pour activer la collecte des informations de statistiques MQI, utilisez la commande MQSC suivante:

```
ALTER QMGR STATMQI(ON)
```

### *Contrôle de la collecte des statistiques de file d'*

Utilisez l'attribut de file d'attente STATQ et l'attribut de gestionnaire de files d'attente STATQ pour contrôler la collecte des informations de statistiques de file d'attente

Vous pouvez activer ou désactiver la collecte des informations de statistiques de file d'attente pour des files d'attente individuelles ou pour plusieurs files d'attente. Pour contrôler des files d'attente individuelles, définissez l'attribut de file d'attente STATQ. Vous activez ou désactivez la collecte des informations de statistiques de file d'attente au niveau du gestionnaire de files d'attente à l'aide de l'attribut de gestionnaire de files d'attente STATQ. Pour toutes les files d'attente pour lesquelles l'attribut de file d'attente STATQ est spécifié avec la valeur QMGR, la collecte des informations statistiques de file d'attente est contrôlée au niveau du gestionnaire de files d'attente.

Les statistiques de file d'attente sont incrémentées uniquement pour les opérations utilisant les descripteurs d'objet IBM MQ MQI qui ont été ouverts après l'activation de la collecte de statistiques.

Les messages de statistiques de file d'attente sont générés uniquement pour les files d'attente pour lesquelles des données statistiques ont été collectées au cours de la période précédente.

La même file d'attente peut comporter plusieurs opérations d'insertion et d'extraction via plusieurs descripteurs d'objet. Certains descripteurs d'objet ont peut-être été ouverts avant l'activation de la collecte de statistiques, mais d'autres ont été ouverts par la suite. Par conséquent, les statistiques de file d'attente peuvent enregistrer l'activité de certaines opérations d'insertion et d'extraction, mais pas toutes.

Pour vous assurer que les statistiques de file d'attente enregistrent l'activité de toutes les applications, vous devez fermer et rouvrir les nouveaux descripteurs d'objet dans la ou les files d'attente que vous surveillez. Le meilleur moyen d'y parvenir est d'arrêter et de redémarrer toutes les applications après l'activation de la collecte de statistiques.

Pour modifier la valeur de l'attribut de file d'attente STATQ, utilisez la commande MQSC ALTER QLOCAL et spécifiez le paramètre STATQ. L'attribut de file d'attente STATQ peut avoir les valeurs suivantes:

#### **Oui**

Les statistiques de file d'attente sont collectées pour chaque connexion au gestionnaire de files d'attente qui ouvre la file d'attente.

#### **Non**

Les informations de statistiques de file d'attente pour cette file d'attente ne sont pas collectées.

#### **QMGR**

La collecte des informations de statistiques de file d'attente pour cette file d'attente est contrôlée en fonction de la valeur de l'attribut de gestionnaire de files d'attente, STATQ. Il s'agit de la valeur par défaut.

Pour modifier la valeur de l'attribut de gestionnaire de files d'attente STATQ, utilisez la commande MQSC ALTER QMGR et spécifiez le paramètre STATQ. L'attribut de gestionnaire de files d'attente STATQ peut avoir les valeurs suivantes:

**Oui**

Les informations de statistiques de file d'attente sont collectées pour les files d'attente dont l'attribut de file d'attente STATQ est défini comme QMGR

**Non**

Les informations de statistiques de file d'attente ne sont pas collectées pour les files dont l'attribut de file d'attente STATQ est défini comme QMGR. Il s'agit de la valeur par défaut.

**Aucun**

La collecte des statistiques de file d'attente est désactivée pour toutes les files d'attente, quel que soit l'attribut de file d'attente STATQ.

Si l'attribut de gestionnaire de files d'attente STATQ est défini sur NONE, la collecte des informations de statistiques de file d'attente est désactivée pour toutes les files d'attente, quel que soit l'attribut de file d'attente STATQ.

Par exemple, pour activer la collecte d'informations statistiques pour la file d'attente, Q1, utilisez la commande MQSC suivante:

```
ALTER QLOCAL(Q1) STATQ(ON)
```

Pour activer la collecte des statistiques pour toutes les files d'attente qui spécifient l'attribut de file d'attente STATQ comme QMGR, utilisez la commande MQSC suivante:

```
ALTER QMGR STATQ(ON)
```

*Contrôle de la collecte des statistiques de canal*

Utilisez l'attribut de canal STATCHL pour contrôler la collecte des informations de statistiques de canal. Vous pouvez également définir des attributs de gestionnaire de files d'attente pour contrôler la collecte d'informations.

Vous pouvez activer ou désactiver la collecte d'informations statistiques sur les canaux pour des canaux individuels ou pour plusieurs canaux. Pour contrôler des canaux individuels, vous devez définir l'attribut de canal STATCHL pour activer ou désactiver la collecte des statistiques de canal. Pour contrôler plusieurs canaux ensemble, vous activez ou désactivez la collecte des statistiques de canal au niveau du gestionnaire de files d'attente à l'aide de l'attribut de gestionnaire de files d'attente STATCHL. Pour tous les canaux dont l'attribut de canal STATCHL est spécifié avec la valeur QMGR, la collecte des informations statistiques de canal est contrôlée au niveau du gestionnaire de files d'attente.

Les canaux émetteurs de cluster définis automatiquement ne sont pas des objets IBM MQ . Par conséquent, ils ne possèdent pas d'attributs de la même manière que les objets de canal. Pour contrôler automatiquement les canaux émetteurs de cluster définis, utilisez l'attribut de gestionnaire de files d'attente STATACLS. Cet attribut détermine si les canaux émetteurs de cluster définis automatiquement dans un gestionnaire de files d'attente sont activés ou désactivés pour la collecte des statistiques de canal.

Vous pouvez définir la collecte des informations de statistiques de canal sur l'un des trois niveaux de surveillance: faible, moyen ou élevé. Vous pouvez définir le niveau de surveillance au niveau de l'objet ou au niveau du gestionnaire de files d'attente. Le choix du niveau à utiliser dépend de votre système. La collecte de données statistiques peut nécessiter des instructions qui sont relativement coûteuses en termes de calcul. Par conséquent, pour réduire l'impact de la collecte d'informations statistiques sur les canaux, les options de surveillance moyenne et faible mesurent un échantillon de données à intervalles réguliers plutôt que de collecter des données en permanence. Le [Tableau 22](#), à la [page 152](#) récapitule les niveaux disponibles avec la collecte d'informations statistiques sur les canaux:

Tableau 22. Niveau de détail de la collecte d'informations statistiques sur les canaux

Niveau	Description	Utilisation
Bas	Mesurez un petit échantillon de données, à intervalles réguliers.	Pour les objets qui traitent un volume élevé de messages.
Moyen	Mesurez un échantillon de données, à intervalles réguliers.	Pour la plupart des objets.
Elevé	Mesurez toutes les données à intervalles réguliers.	Pour les objets qui ne traitent que quelques messages par seconde, sur lesquels les informations les plus récentes sont importantes.

Pour modifier la valeur de l'attribut de canal STATCHL, utilisez la commande MQSC ALTER CHANNEL et spécifiez le paramètre STATCHL.

Pour modifier la valeur de l'attribut de gestionnaire de files d'attente STATCHL, utilisez la commande MQSC ALTER QMGR et spécifiez le paramètre STATCHL.

Pour modifier la valeur de l'attribut de gestionnaire de files d'attente STATACLS, utilisez la commande MQSC ALTER QMGR et spécifiez le paramètre STATACLS.

L'attribut de canal STATCHL peut avoir les valeurs suivantes:

#### **BAS**

Les informations de statistiques de canal sont collectées avec un faible niveau de détail.

#### **MOYENNE**

Les informations de statistiques de canal sont collectées avec un niveau de détail moyen.

#### **ÉLEVÉ**

Les informations de statistiques de canal sont collectées avec un niveau de détail élevé.


#### **Non**

Les informations de statistiques de canal ne sont pas collectées pour ce canal.

#### **QMGR**

L'attribut de canal est défini comme QMGR. La collecte des informations statistiques pour ce canal est contrôlée par la valeur de l'attribut de gestionnaire de files d'attente, STATCHL.

Il s'agit de la valeur par défaut.

 Sur les systèmes z/OS, l'activation de ce paramètre active simplement la collecte de données statistiques, quelle que soit la valeur que vous sélectionnez. La spécification de LOW, MEDIUM ou HIGH n'a pas d'impact sur vos résultats. Vous devez activer ce paramètre pour pouvoir collecter des enregistrements de comptabilité de canal.

L'attribut de gestionnaire de files d'attente STATCHL peut avoir les valeurs suivantes:

#### **BAS**

Les informations de statistiques de canal sont collectées avec un niveau de détail faible pour tous les canaux dont l'attribut de canal STATCHL est défini comme QMGR.

#### **MOYENNE**

Les informations de statistiques de canal sont collectées avec un niveau de détail moyen pour tous les canaux dont l'attribut de canal STATCHL est défini sur QMGR.

#### **ÉLEVÉ**

Les informations de statistiques de canal sont collectées avec un niveau de détail élevé pour tous les canaux dont l'attribut de canal STATCHL est défini comme QMGR.

#### **Non**


Les informations de statistiques de canal ne sont pas collectées pour tous les canaux dont l'attribut de canal STATCHL est défini comme QMGR.



Il s'agit de la valeur par défaut.

### **Aucun**

La collecte des statistiques de canal est désactivée pour tous les canaux, quel que soit l'attribut de canal STATCHL.

 Sur les systèmes z/OS, l'activation de ce paramètre active simplement la collecte de données statistiques, quelle que soit la valeur que vous sélectionnez. La spécification de LOW, MEDIUM ou HIGH n'a pas d'impact sur vos résultats. Vous devez activer ce paramètre pour pouvoir collecter des enregistrements de comptabilité de canal.

L'attribut de gestionnaire de files d'attente, STATACLS, peut avoir les valeurs suivantes:

### **BAS**

Les informations statistiques sont collectées avec un faible niveau de détail pour les canaux émetteurs de cluster définis automatiquement.

### **MOYENNE**

Les informations statistiques sont collectées avec un niveau de détail moyen pour les canaux émetteurs de cluster définis automatiquement.

### **ÉLEVÉ**

Les informations statistiques sont collectées avec un niveau de détail élevé pour les canaux émetteurs de cluster définis automatiquement.


### **Non**

Les informations statistiques ne sont pas collectées pour les canaux émetteurs de cluster définis automatiquement.

### **QMGR**

La collecte des informations statistiques pour les canaux émetteurs de cluster définis automatiquement est contrôlée par la valeur de l'attribut de gestionnaire de files d'attente, STATCHL.

Il s'agit de la valeur par défaut.

 Sur les systèmes z/OS, l'activation de ce paramètre active simplement la collecte de données statistiques, quelle que soit la valeur que vous sélectionnez. La spécification de LOW, MEDIUM ou HIGH n'a pas d'impact sur vos résultats. Vous devez activer ce paramètre pour pouvoir collecter des enregistrements de comptabilité de canal.

Par exemple, pour activer la collecte d'informations statistiques, à un niveau de détail moyen, pour le canal émetteur QM1 . TO . QM2, utilisez la commande MQSC suivante:

```
ALTER CHANNEL(QM1.TO.QM2) CHLTYPE(SDR) STATCHL(MEDIUM)
```

Pour activer la collecte des informations statistiques, à un niveau de détail moyen, pour tous les canaux qui spécifient l'attribut de canal STATCHL comme QMGR, utilisez la commande MQSC suivante:

```
ALTER QMGR STATCHL(MEDIUM)
```

Pour activer la collecte des statistiques, à un niveau de détail moyen, pour tous les canaux émetteurs de cluster définis automatiquement, utilisez la commande MQSC suivante:

```
ALTER QMGR STATACLS(MEDIUM)
```

### *Génération de messages de statistiques*

Les messages de statistiques sont générés à des intervalles configurés et lorsqu'un gestionnaire de files d'attente s'arrête de manière contrôlée.

L'intervalle configuré est contrôlé par l'attribut de gestionnaire de files d'attente STATINT qui indique l'intervalle, en secondes, entre la génération des messages de statistiques. L'intervalle de statistiques par défaut est de 1800 secondes (30 minutes). Pour modifier l'intervalle des statistiques, utilisez la

commande MQSC ALTER QMGR et spécifiez le paramètre **STATINT** . Par exemple, pour remplacer l'intervalle de statistiques par 900 secondes (15 minutes), utilisez la commande MQSC suivante:

```
ALTER QMGR STATINT(900)
```

Pour écrire les données de statistiques actuellement collectées dans la file d'attente de statistiques avant l'expiration de l'intervalle de collecte de statistiques, utilisez la commande MQSC RESET QMGR TYPE (STATISTICS) . L'émission de cette commande entraîne l'écriture des données de statistiques collectées dans la file d'attente de statistiques et le lancement d'un nouvel intervalle de collecte de données de statistiques.

## Affichage des informations de comptabilité et de statistiques

Pour utiliser les informations enregistrées dans les messages de comptabilité et de statistiques, exécutez une application telle que l'exemple de programme **amqsmmon** afin de transformer les informations enregistrées dans un format approprié

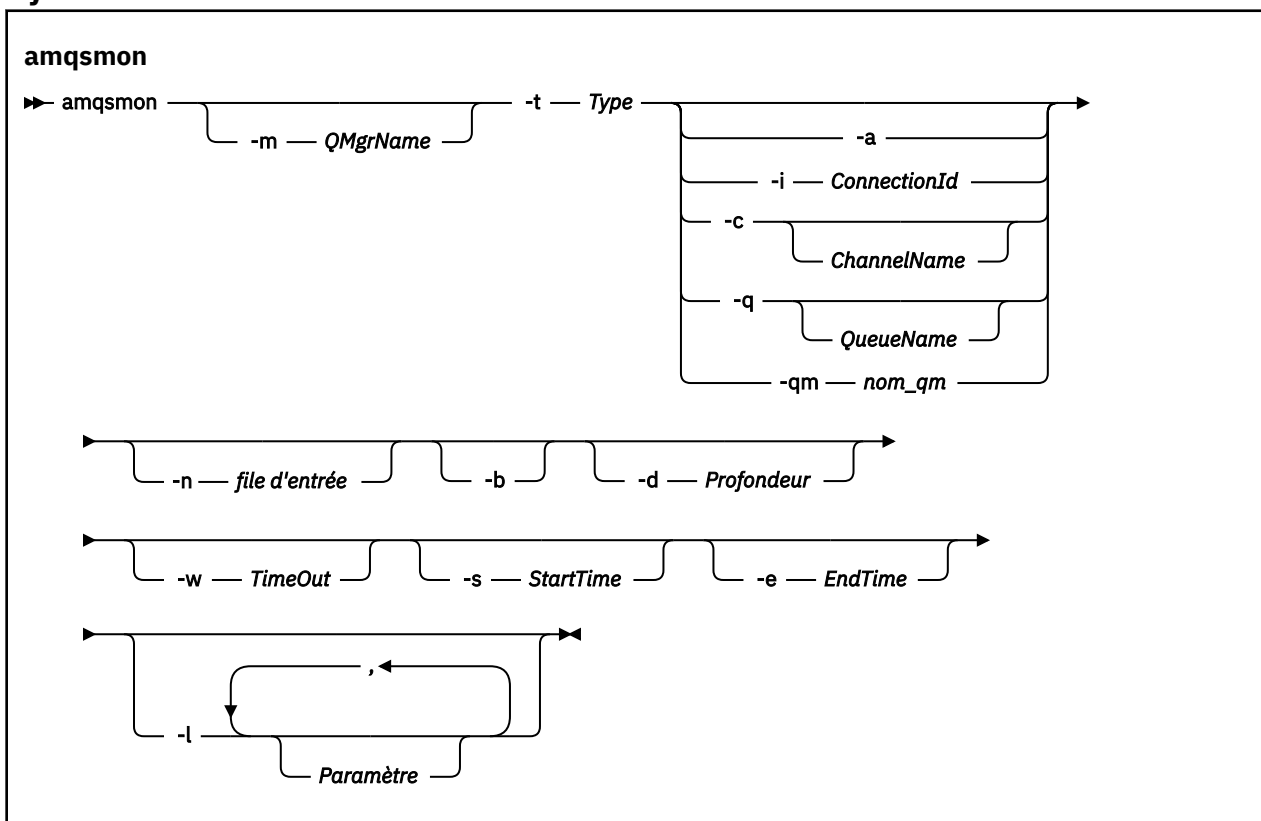
Les messages de comptabilité et de statistiques sont écrits dans les files d'attente de comptabilité et de statistiques du système. **amqsmmon** est un exemple de programme fourni avec IBM MQ qui traite les messages des files d'attente de comptabilité et de statistiques et affiche les informations à l'écran dans un format lisible.

**amqsmmon** étant un exemple de programme, vous pouvez utiliser le code source fourni comme modèle pour écrire votre propre application afin de traiter les messages de comptabilité ou de statistiques, ou modifier le code source **amqsmmon** en fonction de vos besoins.

### ***amqsmmon (affichage des informations de surveillance formatées)***

Utilisez l'exemple de programme **amqsmmon** pour afficher dans un format lisible les informations contenues dans les messages de comptabilité et de statistiques. Le programme **amqsmmon** lit les messages de comptabilité de la file d'attente de comptabilité, SYSTEM.ADMIN.ACCOUNTING.QUEUE. et lit les messages de statistiques de la file d'attente de statistiques, SYSTEM.ADMIN.STATISTICS.QUEUE.

## Syntaxe



## Paramètres requis

### -t type

Type de message à traiter. Spécifiez *Type* de l'une des manières suivantes:

#### comptabilité

Les enregistrements comptables sont traités. Les messages sont lus à partir de la file d'attente système SYSTEM.ADMIN.ACCOUNTING.QUEUE sauf si vous avez utilisé le paramètre `-n` pour sélectionner une file d'attente particulière à partir de laquelle effectuer la lecture.

#### statistiques

Les enregistrements de statistiques sont traités. Les messages sont lus à partir de la file d'attente système SYSTEM.ADMIN.STATISTICS.QUEUE **V 9.4.0** sauf si vous avez utilisé le paramètre `-n` pour sélectionner une file d'attente particulière à partir de laquelle effectuer la lecture.

## Paramètres facultatifs

### -m *NomGestF*

Nom du gestionnaire de files d'attente à partir duquel les messages de comptabilité ou de statistiques doivent être traités.

Si vous ne spécifiez pas ce paramètre, le gestionnaire de files d'attente par défaut est utilisé.

### -a

Traiter les messages contenant des enregistrements MQI uniquement.

Affiche uniquement les enregistrements MQI. Les messages ne contenant pas d'enregistrements MQI seront toujours laissés dans la file d'attente à partir de laquelle ils ont été lus.

### -q *QueueName*

*QueueName* est un paramètre facultatif.

Si <i>QueueName</i> n'est pas fourni:	Affiche uniquement les enregistrements de comptabilité de file d'attente et de statistiques de file d'attente.
Si <i>QueueName</i> est fourni:	Affiche les enregistrements de comptabilité de file d'attente et de statistiques de file d'attente pour la file d'attente spécifiée par <i>QueueName</i> uniquement.
	Si <i>-b</i> n'est pas spécifié, les messages de comptabilité et de statistiques dont proviennent les enregistrements sont supprimés. Etant donné que les messages de comptabilité et de statistiques peuvent également contenir des enregistrements provenant d'autres files d'attente, si <i>-b</i> n'est pas spécifié, les enregistrements invisibles peuvent être supprimés.

### **-c *NomCanal***

*ChannelName* est un paramètre facultatif.

Si <i>ChannelName</i> n'est pas indiqué:	Affiche uniquement les enregistrements de statistiques de canal.
Si <i>ChannelName</i> est fourni:	Affiche les enregistrements de statistiques de canal pour le canal spécifié par <i>ChannelName</i> uniquement.
	Si <i>-b</i> n'est pas spécifié, les messages de statistiques dont proviennent les enregistrements sont supprimés. Etant donné que les messages de statistiques peuvent également contenir des enregistrements provenant d'autres canaux, si <i>-b</i> n'est pas spécifié, les enregistrements invisibles peuvent être supprimés.

Ce paramètre est disponible uniquement lors de l'affichage des messages de statistiques ( *-t statistics* ).

### **V 9.4.0 -qm *nom\_qm***

Ce paramètre est facultatif, mais si vous le spécifiez, vous devez indiquer un nom de gestionnaire de files d'attente

Indique que seuls les messages de comptabilité ou de statistiques générés par le gestionnaire de files d'attente donné doivent être lus à partir de la file d'attente. Généralement, utilisé uniquement si des messages provenant de plusieurs gestionnaires de files d'attente ont été réacheminés vers un gestionnaire de files d'attente central.

### **-i *ConnectionId***

Affiche uniquement les enregistrements liés à l'identificateur de connexion spécifié par *ConnectionId* .

Ce paramètre est disponible uniquement lors de l'affichage des messages de comptabilité ( *-t comptabilité* ).

Si *-b* n'est pas spécifié, les messages de statistiques dont proviennent les enregistrements sont supprimés. Etant donné que les messages de statistiques peuvent également contenir des enregistrements provenant d'autres canaux, si *-b* n'est pas spécifié, les enregistrements invisibles peuvent être supprimés.

### **V 9.4.0 -n *file d'entrée***

File d'attente à partir de laquelle lire les messages de comptabilité ou de statistiques. S'il n'est pas spécifié, la file d'attente par défaut pour le *-t < type >* choisi est utilisée.

### **-b**

Parcourez les messages.

Les messages sont extraits de façon non destructive.

**-d profondeur**

Nombre maximal de messages pouvant être traités.

Si vous ne spécifiez pas ce paramètre, un nombre illimité de messages peut être traité.

**-w délai**

Durée maximale, en secondes, d'attente de la disponibilité d'un message.

Si vous ne spécifiez pas ce paramètre, amqsmn s'arrête une fois qu'il n'y a plus de messages à traiter.

**-s StartTime**

Traiter les messages insérés après le *StartTime* spécifié uniquement.

*StartTime* est spécifié au format yyyy-mm-dd hh.mm.ss. Si une date est spécifiée sans heure, l'heure par défaut est 00.00.00 à la date spécifiée. Les heures sont au format GMT.

Pour connaître l'effet de l'absence de spécification de ce paramètre, voir la [Remarque 1](#).

**-e EndTime**

Traite les messages placés avant l'heure *EndTime* spécifiée uniquement.

*EndTime* est spécifié au format yyyy-mm-dd hh.mm.ss. Si une date est spécifiée sans heure, l'heure par défaut est 23.59.59 à la date spécifiée. Les heures sont au format GMT.

Pour connaître l'effet de l'absence de spécification de ce paramètre, voir la [Remarque 1](#).

**-l Paramètre**

Affiche uniquement les zones sélectionnées à partir des enregistrements traités. *Paramètre* est une liste de valeurs entières séparées par des virgules, chaque valeur entière étant mappée à la constante numérique d'une zone. Voir l' [exemple 5 d'amqsmn](#).

Si vous ne spécifiez pas ce paramètre, toutes les zones disponibles sont affichées.

**Remarque :**

1. Si vous ne spécifiez pas *-s StartTime* ou *-e EndTime*, les messages qui peuvent être traités ne sont pas restreints par l'heure d'insertion.

**Exemples de commande amqsmn (Affichage des informations de surveillance formatées)**

Cette page permet d'afficher des exemples d'exécution de l'exemple de programme amqsmn (Affichage des informations de surveillance formatées)

1. Pour plus d'informations sur les attributs, voir [«Données de message de statistiques de file d'attente»](#), à la page 200 .

La commande suivante affiche tous les messages de statistiques MQI provenant du gestionnaire de files d'attente saturn.queue.manager :

```
amqsmn -m saturn.queue.manager -t statistics -a
```

La sortie de cette commande est la suivante:

```
RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 73, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqCount: [4, 2102, 0, 0, 0, 46, 0, 0, 0, 0, 0, 0, 0]
```

```

InqFailCount: [0, 31, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [26, 1]
PutFailCount: 0
Put1Count: [40, 0]
Put1FailCount: 0
PutBytes: [57064, 12320]
GetCount: [18, 1]
GetBytes: [52, 12320]
GetFailCount: 2254
BrowseCount: [18, 60]
BrowseBytes: [23784, 30760]
BrowseFailCount: 9
CommitCount: 0
CommitFailCount: 0
BackCount: 0
ExpiredMsgCount: 0
PurgeCount: 0

```

2. La commande suivante affiche tous les messages de statistiques de file d'attente pour la file d'attente LOCALQ sur le gestionnaire de files d'attente saturn.queue.manager:

```
amqsmon -m saturn.queue.manager -t statistics -q LOCALQ
```

La sortie de cette commande est la suivante:

```

RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics:
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
  QueueDefinitionType: Local
  QMinDepth: 0
  QMaxDepth: 18
  AverageQueueTime: [29827281, 0]
  PutCount: [26, 0]
  PutFailCount: 0
  Put1Count: [0, 0]
  Put1FailCount: 0
  PutBytes: [88, 0]
  GetCount: [18, 0]
  GetBytes: [52, 0]
  GetFailCount: 0
  BrowseCount: [0, 0]
  BrowseBytes: [0, 0]
  BrowseFailCount: 1
  NonQueuedMsgCount: 0
  ExpiredMsgCount: 0
  PurgedMsgCount: 0

```

3. La commande suivante affiche tous les messages de statistiques enregistrés depuis 15 h 30 le 30 avril 2005 à partir du gestionnaire de files d'attente saturn.queue.manager.

```
amqsmon -m saturn.queue.manager -t statistics -s "2005-04-30 15.30.00"
```

La sortie de cette commande est la suivante:

```

RecordType: MQIStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600

```

```

ConnCount: 23
ConnFailCount: 0
ConnsMax: 8
DiscCount: [17, 0, 0]
OpenCount: [0, 80, 1, 0, 0, 3, 0, 0, 0, 0, 0, 0, 0]
...
RecordType: QueueStatistics
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.02'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.39.02'
CommandLevel: 600
ObjectCount: 3
QueueStatistics: 0
  QueueName: 'LOCALQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...
QueueStatistics: 1
  QueueName: 'SAMPLEQ'
  CreateDate: '2005-03-08'
  CreateTime: '17.07.02'
  QueueType: Predefined
...

```

4. Pour plus d'informations sur les attributs, voir [«Données de message de comptabilité de file d'attente»](#), à la page 177 .

La commande suivante affiche tous les messages de comptabilité enregistrés le 30 avril 2005 à partir du gestionnaire de files d'attente saturn.queue.manager:

```
amqsmon -m saturn.queue.manager -t accounting -s "2005-04-30" -e "2005-04-30"
```

La sortie de cette commande est la suivante:

```

RecordType: MQIAccounting
QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-04-30'
IntervalStartTime: '15.09.29'
IntervalEndDate: '2005-04-30'
IntervalEndTime: '15.09.30'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010a0020'
SeqNumber: 0
ApplicationName: 'amqsput'
ApplicationPid: 8572
ApplicationTid: 1
UserId: 'admin'
ConnDate: '2005-03-16'
ConnTime: '15.09.29'
DiscDate: '2005-03-16'
DiscTime: '15.09.30'
DiscType: Normal
OpenCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
OpenFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseCount: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
CloseFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
PutCount: [1, 0]
PutFailCount: 0
PutBytes: [4, 0]
GetCount: [0, 0]
GetFailCount: 0
GetBytes: [0, 0]
BrowseCount: [0, 0]
BrowseFailCount: 0
BrowseBytes: [0, 0]
CommitCount: 0
CommitFailCount: 0
BackCount: 0
InqCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
InqFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
SetFailCount: [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]
RecordType: MQIAccounting

```

```

QueueManager: 'saturn.queue.manager'
IntervalStartDate: '2005-03-16'
IntervalStartTime: '15.16.22'
IntervalEndDate: '2005-03-16'
IntervalEndTime: '15.16.22'
CommandLevel: 600
ConnectionId: x'414d514354524556312020202020208d0b3742010c0020'
SeqNumber: 0
ApplicationName: 'runmqsc'
ApplicationPid: 8615
ApplicationTid: 1
...

```

5. La commande suivante parcourt la file d'attente de comptabilité et affiche le nom de l'application et l'identificateur de connexion de chaque application pour laquelle des informations de comptabilité MQI sont disponibles:

```
amqsmmon -m saturn.queue.manager -t accounting -b -l 7006,3024
```

La sortie de cette commande est la suivante:

```

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e80322'
ApplicationName: 'WebSphere MQ\bin\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ea0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e90322'
ApplicationName: 's\IBM\MQ_4\bin64\amqsput.exe'
QueueAccounting: 0

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01ef0322'
ApplicationName: 'BM\MQ_4\bin64\MQExplorer.exe'
QueueAccounting: 0
QueueAccounting: 1
QueueAccounting: 2
QueueAccounting: 3
QueueAccounting: 4
QueueAccounting: 5
QueueAccounting: 6
QueueAccounting: 7
QueueAccounting: 8
QueueAccounting: 9

MonitoringType: QueueAccounting
ConnectionId: x'414d5143514d39303520202020202020fcf1855e01e60322'
ApplicationName: 's\IBM\MQ_4\bin64\runmqsc.exe'
QueueAccounting: 0

5 Records Processed

```

Pour plus de détails sur la façon de localiser les variables utilisées dans cette rubrique, voir [«Recherche du mappage pour ApplicationName \(3024\) et ConnectionId \(7006\)»](#), à la page 161 .

Voir [«Recherche du mappage pour AvgTimeOnQ \(703\) et QmaxDepth \(739\)»](#), à la page 161 pour plus de détails sur la façon de localiser les variables si vous avez sélectionné *statistics* pour le paramètre **Type** dans la commande **amqsmmon** .



### Recherche du mappage pour **ApplicationName** (3024) et **ConnectionId** (7006)

Utilisez cette rubrique pour expliquer comment les variables utilisées dans l'exemple sont trouvées pour la surveillance des données comptables.

Récapitulatif :

- 3024 signifie MQCACF\_APPL\_NAME qui correspond à **ApplicationName**
- 7006 signifie MQBACF\_CONNECTION\_ID qui correspond à **ConnectionId**

Pour trouver le mappage, vous devez effectuer deux procédures:

1. Visitez le site «Données de message de comptabilité de file d'attente», à la page 177 pour obtenir une explication des attributs «ApplicationName», à la page 178 et «ConnectionId», à la page 178 lorsque vous avez sélectionné *comptabilité* pour le paramètre **Type** dans la commande **amqsmn** .

Dans chaque cas, recherchez l'attribut **Identifieur**.

**Identifieur** pour **ApplicationName** est MQCACF\_APPL\_NAME et pour **ConnectionId** est MQBACF\_CONNECTION\_ID

2. Recherchez les identificateurs que vous avez trouvés à l'étape «1», à la page 161.

Accédez à la section Constantes et faites défiler jusqu'à la liste MQCACF\_\* (Command format Character Parameter Types) . Localisez MQCACF\_APPL\_NAME et la valeur 3024s'affiche.

De même, recherchez la liste MQBACF\_\* (Command format Byte Parameter Types). Localisez MQBACF\_CONNECTION\_ID et la valeur 7006s'affiche.

### Recherche du mappage pour **AvgTimeOnQ** (703) et **QmaxDepth** (739)

Utilisez cette rubrique pour expliquer comment les variables utilisées dans l'exemple sont trouvées pour les données statistiques de surveillance.

Récapitulatif :

- 703 signifie MQIAMO64\_AVG\_Q\_TIME , qui correspond à **AvgTimeOnQ**
- 739 signifie MQIAMO\_Q\_MAX\_DEPTH qui correspond à **QMaxDepth**

Pour trouver le mappage, vous devez effectuer deux procédures:

1. Visitez le site «Données de message de statistiques de file d'attente», à la page 200 pour obtenir une explication des attributs «AvgTimeOnQ», à la page 203 et «QMaxDepth», à la page 203 lorsque vous avez sélectionné *statistics* pour le paramètre **Type** dans la commande **amqsmn** .

Dans chaque cas, recherchez l'attribut **Identifieur**.

**Identifieur** pour **AvgTimeOnQ** est MQIAMO64\_AVG\_Q\_TIME et pour **QMaxDepth** est MQIAMO\_Q\_MAX\_DEPTH.

2. Recherchez les identificateurs que vous avez trouvés à l'étape «1», à la page 161.

Accédez à la section Constantes et faites défiler l'écran jusqu'à la liste MQIAMO\_\* (Command format Integer Monitoring Parameter Types) . Localisez MQIAMO\_Q\_MAX\_DEPTH et la valeur 739s'affiche.

De même, recherchez la liste MQIAMO64\_\* (Format de commande-Types de paramètre de surveillance des entiers 64 bits) . Localisez MQIAMO64\_AVG\_Q\_TIME et la valeur 703s'affiche.

## Référence des messages de comptabilité et de statistiques

Cette page permet d'obtenir un aperçu du format des messages de comptabilité et de statistiques et des informations renvoyées dans ces messages.

Les messages de comptabilité et de statistiques sont des messages IBM MQ standard contenant un descripteur de message et des données de message. Les données de message contiennent des informations sur les opérations MQI effectuées par les applications IBM MQ ou sur les activités qui se produisent dans un système IBM MQ .

**Descripteur de message**

- Une structure MQMD

**Données de message**

- En-tête PCF (MQCFH)
- Données de message de comptabilité ou de statistiques toujours renvoyées
- Données de message de comptabilité ou de statistiques renvoyées si elles sont disponibles

***Format des messages de comptabilité et de statistiques***

Utilisez cette page comme exemple de structure d'un message de comptabilité MQI

Tableau 23. Structure des messages de comptabilité MQI

Structure MQMD	Structure MQCFH d'en-tête de message de comptabilité	Données de message de comptabilité MQI <sup>1</sup>
Identificateur de structure Version de la structure Options de rapport Type de message Délai d'expiration Code retour Codage CCSID Format de message Priorité de message Persistance Identificateur de message Identificateur de corrélation Nombre d'annulations File de réponses Gestionnaire de file de réponses ID utilisateur Jeton de comptabilité Données sur l'identité de l'application Type d'application Nom d'application Date d'insertion Heure d'insertion Données sur l'origine de l'application Identificateur de groupe Numéro de séquence du message Décalage Indicateurs de message Longueur d'origine	Type de structure Longueur de la structure Version de la structure Identificateur de commande Numéro de séquence du message Options de contrôle Code de fin d'exécution Code raison Nombre de paramètres	Gestionnaire de files d'attente Date de début de l'intervalle Heure de début de l'intervalle Date de fin de l'intervalle Heure de fin d'intervalle Niveau de commande Identificateur de connexion Numéro de séquence Nom d'application Identificateur de processus d'application Identificateur d'unité d'exécution d'application ID utilisateur Date de connexion Durée de connexion Nom de connexion Nom du canal Date de déconnexion Heure de déconnexion Type de déconnexion Nombre ouvertures Nombre d'échecs d'ouverture Nombre de fermetures Nombre d'échecs de fermeture Nombre d'insertions Nombre d'échecs d'insertion Nombre d'insertions Put1 Nombre d'échecs PUT1 Octets insérés Obtenir le nombre Nombre d'échecs d'extraction Octets extraits Nombre de consultations Nombre d'échecs de consultations Octets consultés Nombre de validations Nombre d'échecs de validation Nombre d'annulations Nombre d'interrogations Nombre d'échecs d'interrogations Nombre de définitions Nombre d'échecs de définition
<p><b>Remarque :</b></p> <p>1. Les paramètres affichés sont ceux renvoyés pour un message de comptabilité MQI. Les données de message de comptabilité ou de statistiques réelles dépendent de la catégorie de message.</p>		

## ***MQMD de message de comptabilité et de statistiques (descripteur de message)***

Cette page permet de comprendre les différences entre le descripteur de message des messages de comptabilité et de statistiques et le descripteur de message des messages d'événement.

Les paramètres et les valeurs du descripteur de message du message de comptabilité et de statistiques sont identiques à ceux du descripteur de message des messages d'événement, à l'exception de ce qui suit:

### ***Format***

Description : Nom de format des données de message.  
Type de données : MQCHAR8.  
Valeur : **MQFMT\_ADMIN**  
Message d'administration.

Certains des paramètres contenus dans le descripteur de message du message de comptabilité et de statistiques contiennent des données fixes fournies par le gestionnaire de files d'attente qui a généré le message.

Le MQMD indique également le nom du gestionnaire de files d'attente (tronqué à 28 caractères) qui a inséré le message, ainsi que la date et l'heure auxquelles le message a été inséré dans la file d'attente de comptabilité ou de statistiques.

## ***Données de message dans les messages de comptabilité et de statistiques***

Les données de message dans les messages de comptabilité et de statistiques sont basées sur le format de commande programmable (PCF), qui est utilisé dans les demandes et les réponses de commande PCF. Les données de message dans les messages de comptabilité et de statistiques sont constituées d'un en-tête PCF (MQCFH) et d'un rapport de comptabilité ou de statistiques.

## **Message de comptabilité et de statistiques MQCFH (en-tête PCF)**

L'en-tête de message des messages de comptabilité et de statistiques est une structure MQCFH. Les paramètres et les valeurs de l'en-tête de message du message de comptabilité et de statistiques sont identiques à ceux de l'en-tête de message des messages d'événement, avec les exceptions suivantes:

### ***Command***

Description : Identificateur de commande. Identifie la catégorie de message de comptabilité ou de statistiques.  
Type de données : MQLONG:  
Valeurs : **MQCMD\_ACCOUNTING\_MQI**  
Message de comptabilité MQI.  
**MQCMD\_ACCOUNTING\_Q**  
Message de comptabilité de file d'attente.  
**MQCMD\_STATISTICS\_MQI**  
Message de statistiques MQI.  
**MQCMD\_STATISTICS\_Q**  
Message de statistiques de file d'attente.  
**MQCMD\_STATISTICS\_CHANNEL**  
Message de statistiques de canal.

### ***Version***

Description : Numéro de version de la structure.  
Type de données : MQLONG:

Valeur : **MQCFH\_VERSION\_3**  
Version-3 pour les messages de comptabilité et de statistiques.

## Données de message de comptabilité et de statistiques

Le contenu des données de message de comptabilité et de statistiques dépend de la catégorie du message de comptabilité ou de statistiques, comme suit:

### Message de comptabilité MQI

Les données de message de comptabilité MQI se composent d'un certain nombre de paramètres PCF, mais pas de groupes PCF.

### Message de comptabilité de file d'attente

Les données de message de comptabilité de file d'attente se composent d'un certain nombre de paramètres PCF et sont comprises entre 1 et 100 groupes PCF *QAccountingData* .

### Message de statistiques MQI

Les données de message de statistiques MQI se composent d'un certain nombre de paramètres PCF, mais pas de groupes PCF.

### Message de statistiques de file d'attente

Les données de message de statistiques de file d'attente se composent d'un certain nombre de paramètres PCF et sont comprises entre 1 et 100 groupes PCF *QStatisticsData* .

### Message de statistiques de canal

Les données de message de statistiques de canal se composent d'un certain nombre de paramètres PCF et sont comprises entre 1 et 100 groupes PCF *ChlStatisticsData* .

## Données de message de comptabilité MQI

Cette page permet d'afficher la structure d'un message de comptabilité MQI

Nom du message :	Message de comptabilité MQI.
Plateformes :	Tous, sauf IBM MQ for z/OS.
File d'attente système:	SYSTEM.ADMIN.ACCOUNTING.QUEUE.

### **QueueManager**

Description :	Nom du gestionnaire de files d'attente
Identificateur :	MQCA_Q_MGR_NAME
Type de données :	MQCFST
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH
Renvoyé:	Toujours

### **IntervalStartDate**

Description :	Date de début de la période de surveillance
Identificateur :	DATE MQCAMO_START_DATE
Type de données :	MQCFST
Longueur maximale :	LONGUEUR_DATE_MQ
Renvoyé:	Toujours

### **IntervalStartTime**

Description :	Heure de début de la période de surveillance
---------------	--

Identificateur : HEURE DE DEMARRAGE MQCAMO\_DEMARRAGE  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours

### ***IntervalEndDate***

Description : Date de fin de la période de surveillance  
Identificateur : DATE MQCAMO\_FIN  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Toujours

### ***IntervalEndTime***

Description : Heure de fin de la période de surveillance  
Identificateur : Heure\_END\_MQCAMO  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours

### ***CommandLevel***

Description : Niveau de commande du gestionnaire de files d'attente  
Identificateur : MQIA\_NIVEAU DE COMMANDE  
Type de données : MQCFIN  
Renvoyé: Toujours

### ***ConnectionId***

Description : Identificateur de connexion pour la connexion IBM MQ  
Identificateur : ID\_CONNEXION\_MQBACF  
Type de données : MQCFBS  
Longueur maximale : LONGUEUR\_ID\_CONNEXION\_MQM  
Renvoyé: Toujours

### ***SeqNumber***

Description : Numéro de séquence. Cette valeur est incrémentée pour chaque enregistrement suivant pour les connexions à exécution longue.  
Identificateur : NUMÉRO\_SÉQUENCE\_MQIACF\_XX\_ENCODE\_CASE\_CAPS\_LOCK\_OFF  
Type de données : MQCFIN  
Renvoyé: Toujours

### ***ApplicationName***

Description :	Nom de l'application. Le contenu de cette zone est équivalent au contenu de la zone <i>PutApp1Name</i> dans le descripteur de message.
Identificateur :	NOM_APPL_MQCACF
Type de données :	MQCFST
Longueur maximale :	LONGUEUR_NOM_APPL_MQ_
Renvoyé:	Toujours

### ***ApplicationPid***

Description :	Identificateur de processus du système d'exploitation de l'application
Identificateur :	ID_PROCESSUS_MQIACF_ID
Type de données :	MQCFIN
Renvoyé:	Toujours

### ***ApplicationTid***

Description :	Identificateur d'unité d'exécution IBM MQ de la connexion dans l'application
Identificateur :	ID_UNITÉ_UNITÉ_FILE_MQIACF
Type de données :	MQCFIN
Renvoyé:	Toujours

### ***UserId***

Description :	Contexte d'ID utilisateur de l'application
Identificateur :	MQCACF_IDENTIFICATEUR_UTILISATEUR
Type de données :	MQCFST
Longueur maximale :	LONGUEUR_ID_UTILISATEUR
Renvoyé:	Toujours

### ***ConnDate***

Description :	Date de l'opération MQCONN
Identificateur :	DATE MQCAMO_CONN_DATE
Type de données :	MQCFST
Longueur maximale :	LONGUEUR_TEMPS_MQ_ATTENTE
Renvoyé:	Lorsqu'il est disponible

### ***ConnTime***

Description :	Heure de l'opération MQCONN
Identificateur :	HEURE_CONNEXION MQCAMO_MQ
Type de données :	MQCFST

Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Lorsqu'il est disponible

### **ConnName**

Description : Nom de connexion pour la connexion client  
Identificateur : MQCACH\_NOM\_CONNEXION  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_CONN\_MQ\_  
Renvoyé: Lorsqu'il est disponible

### **ChannelName**

Description : Nom de canal pour la connexion client  
Identificateur : MQCACH\_CHANNEL\_NAME  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_CANAL\_MQ\_CANAL  
Renvoyé: Lorsqu'il est disponible

### **RemoteProduct**

Description : Identificateur de produit distant pour la connexion client, tel qu'affiché dans la zone **RPRODUCT** de DISPLAY CHSATU  
Identificateur : MQCACH\_PRODUI\_T\_DISTANT  
Type de données : MQCFST  
Longueur maximale : MQ\_LONGUEUR\_PRODUI\_T\_DISTANT  
Renvoyé: Lorsqu'il est disponible

### **RemoteVersion**

Description : Version du produit distant pour la connexion client, comme indiqué dans la zone **RVERSION** de DISPLAY CHSTATUS  
Identificateur : VERSION\_DISTANT\_CACHE  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_VERSION\_DISTANT\_MQ\_DISTANTE  
Renvoyé: Lorsqu'il est disponible

### **DiscDate**

Description : Date de l'opération MQDISC  
Identificateur : DATE MQCAMO\_DISC\_DATE  
Type de données : MQCFST



Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Lorsqu'il est disponible

### ***DiscTime***

Description : Heure de l'opération MQDISC  
Identificateur : HEURE-DISC\_MQCAMO\_  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Lorsqu'il est disponible

### ***DiscType***

Description : Type de déconnexion  
Identificateur : TYPE\_DISC\_MQIAMO  
Type de données : MQCFIN  
Valeurs : Les valeurs possibles sont les suivantes:  
**MQDISCONNECT\_NORMAL**  
Demandé par l'application  
**MQDISCONNECT\_IMPLICITE**  
Arrêt anormal de l'application  
**MQDISCONNECT\_Q\_MGR**  
Connexion interrompue par le gestionnaire de files d'attente  
Renvoyé: Lorsqu'il est disponible

### ***OpenCount***

Description : Nombre d'objets ouverts avec succès, soit en émettant directement un appel à MQOPEN, soit en utilisant l'instruction MQPUT1 . Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).  
Identificateur : MQIAMO\_OUVERTURES  
Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible

### ***OpenFailCount***

Description : Nombre de tentatives d'ouverture d'un objet ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).  
Identificateur : Echec de MQIAMO\_OPENS\_FAILED  
Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible

### ***CloseCount***

Description : Nombre d'objets fermés. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).  
Identificateur : MQIAMO\_FERMETURES

Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible

### ***CloseFailCount***

Description : Nombre de tentatives de fermeture d'un objet ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).  
Identificateur : Echec de MQIAMO\_CLOSES\_FAILED  
Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible

### ***PutCount***

Description : Nombre de messages persistants et non persistants correctement insérés dans une file d'attente, à l'exception des messages insérés à l'aide de l'appel MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_INSERTIONS  
Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible

### ***PutFailCount***

Description : Nombre de tentatives d'insertion de message ayant échoué  
Identificateur : MQIAMO\_PUTS\_FAILED  
Type de données : MQCFIN  
Renvoyé: Lorsqu'il est disponible

### ***Put1Count***

Description : Nombre de messages persistants et non persistants correctement insérés dans la file d'attente à l'aide d'appels MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_PUT1S  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***Put1FailCount***

Description : Nombre de tentatives d'insertion de message ayant échoué à l'aide d'appels MQPUT1  
Identificateur : MQIAMO\_PUT1S\_FAILED  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### **PutBytes**

Description : Nombre d'octets écrits à l'aide d'appels d'insertion pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_PUT\_BYTES

Type de données : MQCFIL64

Renvoyé: Lorsqu'il est disponible

### **GetCount**

Description : Nombre d'appels MQGET destructifs réussis pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO\_GETS

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible

### **GetFailCount**

Description : Nombre d'appels MQGET destructifs ayant échoué

Identificateur : Echec de MQIAMO\_GETS\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible

### **GetBytes**

Description : Nombre total d'octets extraits pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_GET\_BYTES

Type de données : MQCFIL64

Renvoyé: Lorsqu'il est disponible

### **BrowseCount**

Description : Nombre d'appels MQGET non destructifs réussis pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO\_NAVIGATIONS

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible

### **BrowseFailCount**

Description : Nombre d'appels MQGET non destructifs ayant échoué

Identificateur : Echec de MQIAMO\_BROWSES\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible

### ***BrowseBytes***

Description : Nombre total d'octets parcourus pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_BROWSE\_BYTES

Type de données : MQCFIL64

Renvoyé: Lorsqu'il est disponible

### ***CommitCount***

Description : Nombre de transactions ayant abouti. Ce nombre inclut les transactions validées implicitement par l'application connectée. Les demandes de validation pour lesquelles il n'y a pas de travail en attente sont incluses dans ce nombre.

Identificateur : MQIAMO\_VALIDATIONS

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible

### ***CommitFailCount***

Description : Nombre de tentatives d'exécution d'une transaction ayant échoué

Identificateur : Echec de MQIAMO\_COMMITS\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible

### ***BackCount***

Description : Nombre d'annulations traitées, y compris les annulations implicites dues à une déconnexion anormale

Identificateur : MQIAMO\_ANNULATIONS

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible

### ***InqCount***

Description : Nombre d'interrogations d'objets ayant abouti. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_INQS

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible

### ***InqFailCount***

Description : Nombre de tentatives d'interrogation d'objet ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : Echec de MQIAMO\_INQS\_FAILED

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible

### **SetCount**

Description : Nombre d'appels MQSET ayant abouti. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_SETS

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible

### **SetFailCount**

Description : Nombre d'appels MQSET ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_SETS\_FAILED

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible

### **SubCountDur**

Description : Nombre de demandes d'abonnement ayant abouti qui ont créé, modifié ou repris des abonnements durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération

0 = Nombre d'abonnements créés

1 = Nombre d'abonnements modifiés

2 = Nombre d'abonnements repris

Identificateur : MQIAMO\_SUBS\_DUR

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible.

### **SubCountNDur**

Description : Nombre de demandes d'abonnement ayant abouti qui ont créé, modifié ou repris des abonnements non durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération

0 = Nombre d'abonnements créés

1 = Nombre d'abonnements modifiés

2 = Nombre d'abonnements repris

Identificateur : MQIAMO\_SUBS\_NDUR

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible.

### **SubFailCount**

Description : Nombre de demandes d'abonnement qui ont échoué.

Identificateur : Echec de MQIAMO\_SUBS\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***UnsubCountDur***

Description : Nombre de demandes de désabonnement réussies pour les abonnements durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération  
0-L'abonnement a été fermé mais n'a pas été supprimé  
1-L'abonnement a été fermé et supprimé

Identificateur : MQIAMO\_UNSUBS\_DUR

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible.

### ***UnsubCountNDur***

Description : Nombre de demandes de désabonnement réussies pour les abonnements durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération  
0-L'abonnement a été fermé mais n'a pas été supprimé  
1-L'abonnement a été fermé et supprimé

Identificateur : MQIAMO\_UNSUBS\_NDUR

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible.

### ***UnsubFailCount***

Description : Nombre de demandes de désabonnement ayant échoué.

Identificateur : MQIAMO\_UNSUBS\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***SubRqCount***

Description : Nombre de demandes MQSUBRQ qui ont abouti.

Identificateur : MQIAMO\_SUBRQS

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***SubRqFailCount***

Description : Nombre de demandes MQSUB ayant échoué.

Identificateur : Echec de MQIAMO\_SUBRQS\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***CBCount***

Description : Nombre de demandes MQCB ayant abouti. Il s'agit d'un tableau de valeurs indexées par le type d'opération

- 0-Un rappel a été créé ou modifié
- 1-Un rappel a été supprimé
- 2-Un rappel a été repris
- 3-Un rappel a été suspendu

Identificateur : MQIAMO\_CBS

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***CBFailCount***

Description : Nombre de demandes MQCB n'ayant pas abouti.

Identificateur : MQIAMO\_CBS\_ECHEC

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***CtlCount***

Description : Nombre de demandes MQCTL ayant abouti. Il s'agit d'un tableau de valeurs indexées par le type d'opération

- 0-La connexion a été démarrée
- 1-La connexion a été arrêtée
- 2-La connexion a été reprise
- 3-La connexion a été interrompue

Identificateur : MQIAMO\_CTLS

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible.

### ***CtlFailCount***

Description : Nombre de demandes MQCTL qui ont échoué.

Identificateur : MQIAMO\_CTLS\_FAILED

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***StatCount***

Description : Nombre de demandes MQSTAT qui ont abouti.

Identificateur : MQIAMO\_STATS.

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### **StatFailCount**

Description : Nombre de demandes MQSTAT qui ont échoué.  
Identificateur : MQIAMO\_STATS\_ECHEC  
Type de données : MQCFIN  
Renvoyé: Lorsqu'il est disponible.

### **PutTopicCount**

Description : Nombre de messages persistants et non persistants correctement insérés dans une rubrique, à l'exception des messages insérés à l'aide de l'appel MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Remarque: Les messages insérés à l'aide d'un alias de file d'attente qui se résolvent en une rubrique sont inclus dans cette valeur.

Identificateur : MQIAMO\_TOPIC\_PUTS  
Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible.

### **PutTopicFailCount**

Description : Nombre de tentatives d'insertion d'un message dans une rubrique qui ont échoué.

Identificateur : MQIAMO\_TOPIC\_PUTS\_FAILED  
Type de données : MQCFIN  
Renvoyé: Lorsqu'il est disponible.

### **Put1TopicCount**

Description : Nombre de messages persistants et non persistants correctement insérés dans une rubrique à l'aide d'appels MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Remarque: Les messages insérés à l'aide d'un alias de file d'attente qui se résolvent en une rubrique sont inclus dans cette valeur.

Identificateur : MQIAMO\_TOPIC\_PUT1S  
Type de données : MQCFIL  
Renvoyé: Lorsqu'il est disponible.

### **Put1TopicFailCount**

Description : Nombre de tentatives infructueuses d'insertion d'un message dans une rubrique à l'aide d'appels MQPUT1 .

Identificateur : MQIAMO\_TOPIC\_PUT1S\_FAILED  
Type de données : MQCFIN  
Renvoyé: Lorsqu'il est disponible.



### ***PutTopicBytes***

Description :	Nombre d'octets écrits à l'aide d'appels d'insertion pour les messages persistants et non persistants qui se résolvent en une opération de publication. Il s'agit du nombre d'octets insérés par l'application et non du nombre d'octets résultant transmis aux abonnés. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO64_TOPIC_PUT_BYTES
Type de données :	MQCFIL64
Renvoyé :	Lorsqu'il est disponible.

### ***Données de message de comptabilité de file d'attente***

Cette page permet d'afficher la structure d'un message de comptabilité de file d'attente

Nom du message : Message de comptabilité de file d'attente.

---

Plateformes : Tous, sauf IBM MQ for z/OS.

---

File d'attente système: SYSTEM.ADMIN.ACCOUNTING.QUEUE.

### ***QueueManager***

Description :	Nom du gestionnaire de files d'attente
Identificateur :	MQCA_Q_MGR_NAME
Type de données :	MQCFST
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH
Renvoyé :	Toujours

### ***IntervalStartDate***

Description :	Date de début de la période de surveillance
Identificateur :	DATE MQCAMO_START_DATE
Type de données :	MQCFST
Longueur maximale :	LONGUEUR_DATE_MQ
Renvoyé :	Toujours

### ***IntervalStartTime***

Description :	Heure de début de la période de surveillance
Identificateur :	HEURE DE DEMARRAGE MQCAMO_DEMARRAGE
Type de données :	MQCFST
Longueur maximale :	LONGUEUR_TEMPS_MQ_ATTENTE
Renvoyé :	Toujours

### ***IntervalEndDate***

Description :	Date de fin de la période de surveillance
Identificateur :	DATE MQCAMO_FIN

Type de données : MQCFST  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Toujours

### ***IntervalEndTime***

Description : Heure de fin de la période de surveillance  
Identificateur : Heure\_END\_MQCAMO  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours

### ***CommandLevel***

Description : Niveau de commande du gestionnaire de files d'attente  
Identificateur : MQIA\_NIVEAU DE COMMANDE  
Type de données : MQCFIN  
Renvoyé: Toujours

### ***ConnectionId***

Description : Identificateur de connexion pour la connexion IBM MQ  
Identificateur : ID\_CONNEXION\_MQBACF  
Type de données : MQCFBS  
Longueur maximale : LONGUEUR\_ID\_CONNEXION\_MQM  
Renvoyé: Toujours

### ***SeqNumber***

Description : Numéro de séquence. Cette valeur est incrémentée pour chaque enregistrement suivant pour les connexions à exécution longue.  
Identificateur : NUMÉRO\_SÉQUENCE\_MQIACF\_XX\_ENCODE\_CASE\_CAPS\_LOCK\_OFF  
Type de données : MQCFIN  
Renvoyé: Toujours

### ***ApplicationName***

Description : Nom de l'application. Le contenu de cette zone est équivalent au contenu de la zone de nom PutAppl dans le descripteur de message.  
Identificateur : NOM\_APPL\_MQCACF  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_APPL\_MQ\_  
Renvoyé: Toujours

### **ApplicationPid**

Description : Identificateur de processus du système d'exploitation de l'application  
Identificateur : ID\_PROCESSUS\_MQIACF\_ID  
Type de données : MQCFIN  
Renvoyé: Toujours

### **ApplicationTid**

Description : Identificateur d'unité d'exécution IBM MQ de la connexion dans l'application  
Identificateur : ID\_UNITÉ\_UNITÉ\_FILE\_MQIACF  
Type de données : MQCFIN  
Renvoyé: Toujours

### **UserId**

Description : Contexte d'ID utilisateur de l'application  
Identificateur : MQCACF\_IDENTIFICATEUR\_UTILISATEUR  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_ID\_UTILISATEUR  
Renvoyé: Toujours

### **ChannelName**

Description : Nom de canal pour la connexion client  
Identificateur : MQCACH\_CHANNEL\_NAME  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_CANAL\_MQ\_CANAL  
Renvoyé: Lorsqu'il est disponible

### **ConnName**

Description : Nom de connexion pour la connexion client  
Identificateur : MQCACH\_NOM\_CONNEXION  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_CONN\_MQ\_  
Renvoyé: Lorsqu'il est disponible

### **ObjectCount**

Description : Nombre de files d'attente consultées dans l'intervalle pour lequel des données de comptabilité ont été enregistrées. Cette valeur est définie sur le nombre de groupes PCF *QAccountingData* contenus dans le message.  
Identificateur : NOMBRE\_OBJETS\_MQIAMO\_OBJET  
Type de données : MQCFIN

Renvoyé: Toujours

### **QAccountingData**

Description : Paramètres groupés spécifiant les détails de comptabilité d'une file d'attente

Identificateur : MQGACF\_Q\_DONNEES\_COMPTE

Type de données : MQCFGR

Paramètres du groupe:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- OpenCount*
- OpenDate*
- OpenTime*
- CloseDate*
- CloseTime*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- PutMinBytes*
- PutMaxBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- GetMinBytes*
- GetMaxBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- BrowseMinBytes*
- BrowseMaxBytes*
- TimeOnQMin*
- TimeOnQAvg*
- TimeOnQMax*

Renvoyé: Toujours

### **QName**

Description : Nom de la file d'attente

Identificateur : NOM\_Q\_MQCA

Type de données : MQCFST

Inclus dans le groupe PCF: *QAccountingData*

Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_

Renvoyé: Lorsqu'il est disponible

### **CreateDate**

Description : Date de création de la file d'attente  
Identificateur : DATE DE CREATION\_MQCA  
Type de données : MQCFST  
Inclus dans le groupe PCF: *QAccountingData*  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Lorsqu'il est disponible

### **CreateTime**

Description : Heure de création de la file d'attente  
Identificateur : HEURE DE CREATION\_MQCA  
Type de données : MQCFST  
Inclus dans le groupe PCF: *QAccountingData*  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Lorsqu'il est disponible

### **QType**

Description : Type de la file d'attente  
Identificateur : MQIA\_Q\_TYPE  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QAccountingData*  
Valeur : MQQT\_LOCAL  
Renvoyé: Lorsqu'il est disponible

### **QDefinitionType**

Description : Type de définition de file d'attente  
Identificateur : MQIA\_DEFINITION\_TYPE  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QAccountingData*  
Valeurs : Les valeurs possibles sont les suivantes :  
**MQQDT\_PRÉDÉFINI**  
**MQQDT\_PERMANENT\_DYNAMIQUE**  
**MQQDT\_TEMPORARY\_DYNAMIQUE**  
Renvoyé: Lorsqu'il est disponible

### **OpenCount**

Description :	Nombre de fois où cette file d'attente a été ouverte par l'application dans cet intervalle, en émettant directement un appel à MQOPEN ou en utilisant l'instruction MQPUT1 .
Identificateur :	MQIAMO_OUVERTURES
Type de données :	MQCFIL
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **OpenDate**

Description :	Date à laquelle la file d'attente a été ouverte pour la première fois au cours de cet intervalle d'enregistrement. Si la file d'attente était déjà ouverte au début de cet intervalle, cette valeur reflète la date d'ouverture initiale de la file d'attente.
Identificateur :	DATE MQCAMO_OPEN_DATE
Type de données :	MQCFST
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **OpenTime**

Description :	Heure à laquelle la file d'attente a été ouverte pour la première fois au cours de cet intervalle d'enregistrement. Si la file d'attente était déjà ouverte au début de cet intervalle, cette valeur reflète l'heure à laquelle la file d'attente a été ouverte à l'origine.
Identificateur :	HEURE DE L'OPEN_MQCAMO
Type de données :	MQCFST
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **CloseDate**

Description :	Date de la dernière fermeture de la file d'attente dans cet intervalle d'enregistrement. Si la file d'attente est toujours ouverte, la valeur n'est pas renvoyée.
Identificateur :	DATE MQCAMO_CLOSE_DATE
Type de données :	MQCFST
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **CloseTime**

Description :	Heure de la fermeture finale de la file d'attente dans cet intervalle d'enregistrement. Si la file d'attente est toujours ouverte, la valeur n'est pas renvoyée.
---------------	--

Identificateur : HEURE MQCAMO\_CLOSE  
Type de données : MQCFST  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***PutCount***

Description : Nombre de messages persistants et non persistants insérés dans la file d'attente, à l'exception des appels MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_INSERTIONS  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***PutFailCount***

Description : Nombre de tentatives d'insertion de message ayant échoué, à l'exception des appels MQPUT1  
Identificateur : MQIAMO\_PUTS\_FAILED  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***Put1Count***

Description : Nombre de messages persistants et non persistants correctement insérés dans la file d'attente à l'aide d'appels MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_PUT1S  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***Put1FailCount***

Description : Nombre de tentatives d'insertion de message ayant échoué à l'aide d'appels MQPUT1  
Identificateur : MQIAMO\_PUT1S\_FAILED  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### **PutBytes**

Description :	Nombre total d'octets insérés pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO64_PUT_BYTES
Type de données :	MQCFIL64
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **PutMinBytes**

Description :	Taille minimale des messages persistants et non persistants placés dans la file d'attente. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO_PUT_MIN_BYTES
Type de données :	MQCFIL
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **PutMaxBytes**

Description :	Taille maximale des messages persistants et non persistants placés dans la file d'attente. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO_PUT_MAX_BYTES
Type de données :	MQCFIL
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **GeneratedMsgCount**

Description :	Nombre de messages générés. Les messages générés sont <ul style="list-style-type: none"><li>• Événements de longueur élevée de file d'attente</li><li>• Événements de longueur faible de file d'attente</li></ul>
Identificateur :	MQIAMO_MSGS
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **GetCount**

Description :	Nombre d'appels MQGET destructifs réussis pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
---------------	--



Identificateur : MQIAMO\_GETS  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***GetFailCount***

Description : Nombre d'appels MQGET destructifs ayant échoué  
Identificateur : Echec de MQIAMO\_GETS\_FAILED  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***GetBytes***

Description : Nombre d'octets lus dans les appels MQGET destructifs pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO64\_GET\_BYTES  
Type de données : MQCFIL64  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***GetMinBytes***

Description : Taille du plus petit message persistant et non persistant extrait de la file d'attente. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_GET\_MIN\_BYTES  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***GetMaxBytes***

Description : Taille du plus grand message persistant et non persistant extrait de la file d'attente. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_GET\_MAX\_BYTES  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### **BrowseCount**

Description :	Nombre d'appels MQGET non destructifs réussis pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO_NAVIGATIONS
Type de données :	MQCFIL
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **BrowseFailCount**

Description :	Nombre d'appels MQGET non destructifs ayant échoué
Identificateur :	Echec de MQIAMO_BROWSES_FAILED
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **BrowseBytes**

Description :	Nombre d'octets lus dans les appels MQGET non destructifs ayant renvoyé des messages persistants
Identificateur :	MQIAMO64_BROWSE_BYTES
Type de données :	MQCFIL64
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **BrowseMinBytes**

Description :	Taille du plus petit message persistant et non persistant consulté à partir de la file d'attente. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO_BROWSE_MIN_BYTES
Type de données :	MQCFIL
Inclus dans le groupe PCF:	<i>QAccountingData</i>
Renvoyé:	Lorsqu'il est disponible

### **BrowseMaxBytes**

Description :	Taille du plus grand message persistant et non persistant consulté dans la file d'attente. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO_BROWSE_MAX_BYTES
Type de données :	MQCFIL

Inclus dans le groupe PCF: *QAccountingData*  
Renvoyé: Lorsqu'il est disponible

### ***TimeOnQMin***

Description : Durée la plus courte pendant laquelle un message persistant et non persistant est resté dans la file d'attente avant d'être extrait de façon destructive, en microsecondes. Pour les messages extraits sous le point de synchronisation, cette valeur n'inclut pas le temps avant la validation de l'opération d'extraction. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_Q\_TIME\_MIN

Type de données : MQCFIL64

Inclus dans le groupe PCF: *QAccountingData*

Renvoyé: Lorsqu'il est disponible

### ***TimeOnQAvg***

Description : Durée moyenne pendant laquelle un message persistant et non persistant est resté dans la file d'attente avant d'être extrait de façon destructive, en microsecondes. Pour les messages extraits sous le point de synchronisation, cette valeur n'inclut pas le temps avant la validation de l'opération d'extraction. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_Q\_TIME\_AVG

Type de données : MQCFIL64

Inclus dans le groupe PCF: *QAccountingData*

Renvoyé: Lorsqu'il est disponible

### ***TimeOnQMax***

Description : Durée la plus longue pendant laquelle un message persistant et non persistant est resté dans la file d'attente avant d'être extrait de façon destructive, en microsecondes. Pour les messages extraits sous le point de synchronisation, cette valeur n'inclut pas le temps avant la validation de l'opération d'extraction. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_Q\_TIME\_MAX

Type de données : MQCFIL64

Inclus dans le groupe PCF: *QAccountingData*

Renvoyé: Lorsqu'il est disponible

### **CBCount**

Description :	Nombre de demandes MQCB ayant abouti. Il s'agit d'un tableau de valeurs indexées par le type d'opération 0-Un rappel a été créé ou modifié 1-Un rappel a été supprimé 2-Un rappel a été repris 3-Un rappel a été suspendu
Identificateur :	MQIAMO_CBS
Type de données :	MQCFIN
Renvoyé:	Lorsqu'il est disponible.

### **CBFailCount**

Description :	Nombre de demandes MQCB n'ayant pas abouti.
Identificateur :	MQIAMO_CBS_ECHEC
Type de données :	MQCFIN
Renvoyé:	Lorsqu'il est disponible.

### **Données de message de statistiques MQI**

Cette page permet d'afficher la structure d'un message de statistiques MQI

Nom du message :	Message de statistiques MQI.
Plateformes :	Tous, sauf IBM MQ for z/OS.
File d'attente système:	SYSTEM.ADMIN.STATISTICS.QUEUE.

### **QueueManager**

Description :	Nom du gestionnaire de files d'attente.
Identificateur :	MQCA_Q_MGR_NAME.
Type de données :	MQCFST.
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH.
Renvoyé:	Toujours.

### **IntervalStartDate**

Description :	Date de début de la période de surveillance.
Identificateur :	MQCAMO_START_DATE.
Type de données :	MQCFST.
Longueur maximale :	LONGUEUR_DATE_MQ
Renvoyé:	Toujours.

### **IntervalStartTime**

Description :	Heure de début de la période de surveillance.
Identificateur :	MQCAMO_START_TIME.

Type de données : MQCFST.  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours.

### ***IntervalEndDate***

Description : Date à la fin de la période de surveillance.  
Identificateur : MQCAMO\_END\_DATE:  
Type de données : MQCFST.  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Toujours.

### ***IntervalEndTime***

Description : Heure à la fin de la période de surveillance.  
Identificateur : MQCAMO\_END\_TIME.  
Type de données : MQCFST.  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours.

### ***CommandLevel***

Description : Niveau de commande du gestionnaire de files d'attente.  
Identificateur : MQIA\_COMMAND\_LEVEL.  
Type de données : MQCFIN.  
Renvoyé: Toujours.

### ***ConnCount***

Description : Nombre de connexions réussies au gestionnaire de files d'attente.  
Identificateur : MQIAMO\_CONNS.  
Type de données : MQCFIN.  
Renvoyé: Lorsqu'il est disponible.

### ***ConnFailCount***

Description : Nombre de tentatives de connexion qui ont échoué.  
Identificateur : Echec de MQIAMO\_CONNS\_FAILED.  
Type de données : MQCFIN.  
Renvoyé: Lorsqu'il est disponible.

### ***ConnsMax***

Description : Nombre maximal de connexions simultanées pendant l'intervalle d'enregistrement.  
Identificateur : MQIAMO\_CONNS\_MAX:

Type de données : MQCFIN.  
Renvoyé: Lorsqu'il est disponible.

### ***DiscCount***

Description : Nombre de déconnexions du gestionnaire de files d'attente. Il s'agit d'un tableau d'entiers, indexé par les constantes suivantes:

- MQDISCONNECT\_NORMAL
- MQDISCONNECT\_IMPLICITE
- MQDISCONNECT\_Q\_MGR

Identificateur : MQIAMO\_DISQUES.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***OpenCount***

Description : Nombre d'objets ouverts avec succès, soit en émettant directement un appel à MQOPEN, soit en utilisant l'instruction MQPUT1 . Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_OPENS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***OpenFailCount***

Description : Nombre de tentatives d'ouverture d'objet ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_OPENS\_FAILED.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***CloseCount***

Description : Nombre d'objets dont la fermeture a abouti. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_CLOSE.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***CloseFailCount***

Description : Nombre de tentatives de fermeture d'objet ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_CLOSES\_FAILED.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***InqCount***

Description : Le nombre d'objets a été demandé avec succès. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_INQS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***InqFailCount***

Description : Nombre de tentatives d'interrogation d'objet ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_INQS\_FAILED.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***SetCount***

Description : Nombre d'objets dont la mise à jour a abouti (SET). Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : MQIAMO\_SETS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***SetFailCount***

Description : Nombre de tentatives SET ayant échoué. Ce paramètre est une liste d'entiers indexée par type d'objet. Voir [Note de référence 1](#).

Identificateur : Echec de MQIAMO\_SETS\_FAILED.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***PutCount***

Description : Nombre de messages persistants et non persistants insérés dans une file d'attente, à l'exception des demandes MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO\_PUTS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***PutFailCount***

Description : Nombre de tentatives d'insertion de messages qui ont échoué.

Identificateur : MQIAMO\_PUTS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***Put1Count***

Description : Nombre de messages persistants et non persistants correctement insérés dans une file d'attente à l'aide de demandes MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO\_PUT1S.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***Put1FailCount***

Description : Nombre de tentatives d'insertion d'un message persistant et non persistant dans une file d'attente à l'aide des demandes MQPUT1. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO\_PUT1S\_FAILED.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***PutBytes***

Description : Nombre d'octets pour les messages persistants et non persistants écrits à l'aide de demandes d'insertion. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO64\_PUT\_BYTES.

Type de données : MQCFIL64.

Renvoyé: Lorsqu'il est disponible.

### ***GetCount***

Description : Nombre de demandes d'extraction destructives réussies pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO\_GETS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***GetFailCount***

Description : Nombre de demandes d'insertion destructrices qui ont échoué.

Identificateur : Echec de MQIAMO\_GETS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***GetBytes***

Description : Nombre d'octets lus dans les demandes d'extraction destructive pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO64\_GET\_BYTES.

Type de données : MQCFIL64.



Renvoyé: Lorsqu'il est disponible.

### ***BrowseCount***

Description : Nombre de demandes d'extraction non destructives réussies pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO\_BROWSES.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***BrowseFailCount***

Description : Nombre de demandes d'insertion non destructrices qui ont échoué.

Identificateur : MQIAMO\_BROWSES\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***BrowseBytes***

Description : Nombre d'octets lus dans les demandes d'extraction non destructives pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#)

Identificateur : MQIAMO64\_BROWSE\_BYTES.

Type de données : MQCFIL64.

Renvoyé: Lorsqu'il est disponible.

### ***CommitCount***

Description : Nombre de transactions ayant abouti. Ce nombre inclut les transactions validées implicitement par la déconnexion de l'application et les demandes de validation pour lesquelles il n'y a pas de travail en attente.

Identificateur : MQIAMO\_COMMITS.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***CommitFailCount***

Description : Nombre de tentatives d'exécution d'une transaction qui ont échoué.

Identificateur : MQIAMO\_COMMITS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***BackCount***

Description : Nombre d'annulations traitées, y compris l'annulation implicite lors d'une déconnexion anormale.

Identificateur : MQIAMO\_BACKOUTS.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***ExpiredMsgCount***

Description : Nombre de messages persistants et non persistants qui ont été supprimés au moment de leur expiration avant de pouvoir être extraits.

Identificateur : MQIAMO\_MSGS\_EXPIRED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***PurgeCount***

Description : Nombre de fois où la file d'attente a été effacée.

Identificateur : MQIAMO\_MSGS\_PURGED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***SubCountDur***

Description : Nombre de demandes d'abonnement réussies qui ont créé, modifié ou repris des abonnements durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération

0 = Nombre d'abonnements créés

1 = Nombre d'abonnements modifiés

2 = Nombre d'abonnements repris

Identificateur : MQIAMO\_SUBS\_DUR.

Type de données : MQCFIL

Renvoyé: Lorsqu'il est disponible.

### ***SubCountNDur***

Description : Nombre de demandes d'abonnement ayant abouti qui ont créé, modifié ou repris des abonnements non durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération

0 = Nombre d'abonnements créés

1 = Nombre d'abonnements modifiés

2 = Nombre d'abonnements repris

Identificateur : MQIAMO\_SUBS\_NDUR.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***SubFailCount***

Description : Nombre de demandes d'abonnement qui ont échoué.

Identificateur : MQIAMO\_SUBS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***UnsubCountDur***

Description : Nombre de demandes de désabonnement réussies pour les abonnements durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération  
0-L'abonnement a été fermé mais n'a pas été supprimé  
1-L'abonnement a été fermé et supprimé

Identificateur : MQIAMO\_UNSUBS\_DUR.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***UnsubCountNDur***

Description : Nombre de demandes de désabonnement ayant abouti pour des abonnements non durables. Il s'agit d'un tableau de valeurs indexées par le type d'opération  
0-L'abonnement a été fermé mais n'a pas été supprimé  
1-L'abonnement a été fermé et supprimé

Identificateur : MQIAMO\_UNSUBS\_NDUR.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***UnsubFailCount***

Description : Nombre de demandes de désabonnement qui ont échoué.

Identificateur : MQIAMO\_UNSUBS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***SubRqCount***

Description : Nombre de demandes MQSUBRQ qui ont abouti.

Identificateur : MQIAMO\_SUBRQS

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

### ***SubRqFailCount***

Description : Nombre de demandes MQSUBRQ qui ont échoué.

Identificateur : MQIAMO\_SUBRQS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***CBCount***

Description : Nombre de demandes MQCB ayant abouti. Il s'agit d'un tableau de valeurs indexées par le type d'opération

- 0-Un rappel a été créé ou modifié
- 1-Un rappel a été supprimé
- 2-Un rappel a été repris
- 3-Un rappel a été suspendu

Identificateur : MQIAMO\_CBS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***CBFailCount***

Description : Nombre de demandes MQCB n'ayant pas abouti.

Identificateur : MQIAMO\_CBS\_ECHEC.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***CtlCount***

Description : Nombre de demandes MQCTL ayant abouti. Il s'agit d'un tableau de valeurs indexées par le type d'opération:

- 0-La connexion a été démarrée
- 1-La connexion a été arrêtée
- 2-La connexion a été reprise
- 3-La connexion a été interrompue

Identificateur : MQIAMO\_CTLS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***CtlFailCount***

Description : Nombre de demandes MQCTL qui ont échoué.

Identificateur : MQIAMO\_CTLS\_FAILED.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***StatCount***

Description : Nombre de demandes MQSTAT qui ont abouti.

Identificateur : MQIAMO\_STATS.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***StatFailCount***

Description : Nombre de demandes MQSTAT qui ont échoué.  
Identificateur : MQIAMO\_STATS\_FAILED.  
Type de données : MQCFIN.  
Renvoyé: Lorsqu'il est disponible.

### ***SubCountDurHighWater***

Description : Cote d'alerte haute sur le nombre d'abonnements durables pendant l'intervalle de temps. Il s'agit d'un tableau de valeurs indexées par SUBTYPE  
0-cote d'alerte haute pour tous les abonnements durables dans le système  
1-La cote d'alerte haute pour les abonnements d'application durables (MQSUBTYPE\_API)  
2-La cote d'alerte haute pour l'abonnement d'administration durable (MQSUBTYPE\_ADMIN)  
3-La cote d'alerte haute pour les abonnements de proxy durables (MQSUBTYPE\_PROXY)

Identificateur : MQIAMO\_SUB\_DUR\_HIGHWATER  
Type de données : MQCFIL.  
Renvoyé: Lorsqu'il est disponible.

### ***SubCountDurLowWater***

Description : Cote d'alerte basse sur le nombre d'abonnements durables au cours de l'intervalle de temps. Il s'agit d'un tableau de valeurs indexées par SUBTYPE.  
0-cote d'alerte basse pour tous les abonnements durables dans le système  
1-La cote d'alerte basse pour les abonnements d'application durables (MQSUBTYPE\_API)  
2-La cote d'alerte basse pour les abonnements d'administration durables (MQSUBTYPE\_ADMIN)  
3-La cote d'alerte basse pour les abonnements de proxy durables (MQSUBTYPE\_PROXY)

Identificateur : MQIAMO\_SUB\_DUR\_LOWWATER  
Type de données : MQCFIL.  
Renvoyé: Lorsqu'il est disponible.

### ***SubCountNDurHighWater***

Description : Cote d'alerte haute sur le nombre d'abonnements non durables pendant l'intervalle de temps. Il s'agit d'un tableau de valeurs indexées par SUBTYPE

0-La cote d'alerte haute pour tous les abonnements non durables dans le système

1-Valeur la plus élevée pour les abonnements d'application non durables (MQSUBTYPE\_API)

2-La cote d'alerte haute pour l'abonnement admin non durable (MQSUBTYPE\_ADMIN)

3-La cote d'alerte haute pour les abonnements de proxy non durables (MQSUBTYPE\_PROXY)

Identificateur : MQIAMO\_SUB\_NDUR\_HIGHWATER

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***SubCountNDurLowWater***

Description : Cote d'alerte basse sur le nombre d'abonnements non durables au cours de l'intervalle de temps. Il s'agit d'un tableau de valeurs indexées par SUBTYPE.

0-La cote d'alerte basse pour tous les abonnements non durables dans le système

1-La cote d'alerte basse pour les abonnements d'application non durables (MQSUBTYPE\_API)

2-La cote d'alerte basse pour les abonnements d'administration non durables (MQSUBTYPE\_ADMIN)

3-La cote d'alerte basse pour les abonnements de proxy non durables (MQSUBTYPE\_PROXY)

Identificateur : MQIAMO\_SUB\_NDUR\_LOWWATER

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***PutTopicCount***

Description : Nombre de messages persistants et non persistants correctement insérés dans une rubrique, à l'exception des messages insérés à l'aide de l'appel MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Remarque: Les messages insérés à l'aide d'un alias de file d'attente qui se résolvent en une rubrique sont inclus dans cette valeur.

Identificateur : MQIAMO\_TOPIC\_PUTS.

Type de données : MQCFIL.

Renvoyé: Lorsqu'il est disponible.

### ***PutTopicFailCount***

Description : Nombre de tentatives d'insertion d'un message dans une rubrique qui ont échoué.

Identificateur : MQIAMO\_TOPIC\_PUTS\_FAILED.  
Type de données : MQCFIN.  
Renvoyé: Lorsqu'il est disponible.

### ***Put1TopicCount***

Description : Nombre de messages persistants et non persistants correctement insérés dans une rubrique à l'aide d'appels MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Remarque: Les messages insérés à l'aide d'un alias de file d'attente qui se résolvent en une rubrique sont inclus dans cette valeur.

Identificateur : MQIAMO\_TOPIC\_PUT1S.  
Type de données : MQCFIL.  
Renvoyé: Lorsqu'il est disponible.

### ***Put1TopicFailCount***

Description : Nombre de tentatives infructueuses d'insertion d'un message dans une rubrique à l'aide d'appels MQPUT1 .

Identificateur : MQIAMO\_TOPIC\_PUT1S\_FAILED.  
Type de données : MQCFIN.  
Renvoyé: Lorsqu'il est disponible.

### ***PutTopicBytes***

Description : Nombre d'octets écrits à l'aide d'appels d'insertion pour les messages persistants et non persistants qui se résolvent en une opération de publication. Il s'agit du nombre d'octets insérés par l'application et non du nombre d'octets résultant transmis aux abonnés. Voir PublishMsg pour cette valeur. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_TOPIC\_PUT\_BYTES.  
Type de données : MQCFIL64.  
Renvoyé: Lorsqu'il est disponible.

### ***PublishMsgCount***

Description : Nombre de messages distribués aux abonnements dans l'intervalle de temps. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_PUBLISH\_MSG\_COUNT  
Type de données : MQCFIL.  
Renvoyé: Lorsqu'il est disponible.

### ***PublishMsgBytes***

Description : Nombre d'octets distribués aux abonnements dans l'intervalle de temps. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO64\_PUBLISH\_MSG\_BYTES

Type de données : MQCFIL64.  
Renvoyé: Lorsqu'il est disponible.

### ***Données de message de statistiques de file d'attente***

Cette page permet d'afficher la structure d'un message de statistiques de file d'attente

Nom du message : Message de statistiques de file d'attente.

---

Plateformes : Tous, sauf IBM MQ for z/OS.

---

File d'attente système: SYSTEM.ADMIN.STATISTICS.QUEUE.

#### ***QueueManager***

Description : Nom du gestionnaire de files d'attente  
Identificateur : MQCA\_Q\_MGR\_NAME  
Type de données : MQCFST  
Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH  
Renvoyé: Toujours

#### ***IntervalStartDate***

Description : Date de début de la période de surveillance  
Identificateur : DATE MQCAMO\_START\_DATE  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Toujours

#### ***IntervalStartTime***

Description : Heure au début de la période de surveillance  
Identificateur : HEURE DE DEMARRAGE MQCAMO\_DEMARRAGE  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours

#### ***IntervalEndDate***

Description : Date à la fin de la période de surveillance  
Identificateur : DATE MQCAMO\_FIN  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Toujours

#### ***IntervalEndTime***

Description : Heure à la fin de la période de surveillance



Identificateur : Heure\_END\_MQCAMO  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours

### **CommandLevel**

Description : Niveau de commande du gestionnaire de files d'attente  
Identificateur : MQIA\_NIVEAU DE COMMANDE  
Type de données : MQCFIN  
Renvoyé: Toujours

### **ObjectCount**

Description : Nombre d'objets de file d'attente consultés dans l'intervalle pour lequel des données statistiques ont été enregistrées. Cette valeur est définie sur le nombre de groupes PCF QStatisticsData contenus dans le message.  
Identificateur : NOMBRE\_OBJETS\_MQIAMO\_OBJET  
Type de données : MQCFIN  
Renvoyé: Toujours

### **QStatisticsData**

Description : Paramètres groupés spécifiant les détails des statistiques d'une file d'attente  
Identificateur : MQGACF\_Q\_STATISTICS\_DATA  
Type de données : MQCFGR

Paramètres du groupe:

- QName*
- CreateDate*
- CreateTime*
- QType*
- QDefinitionType*
- QMinDepth*
- QMaxDepth*
- AvgTimeOnQ*
- PutCount*
- PutFailCount*
- Put1Count*
- Put1FailCount*
- PutBytes*
- GetCount*
- GetFailCount*
- GetBytes*
- BrowseCount*
- BrowseFailCount*
- BrowseBytes*
- NonQueuedMsgCount*
- ExpiredMsgCount*
- PurgeCount*

Renvoyé: Toujours

### ***QName***

Description : Nom de la file d'attente  
Identificateur : NOM\_Q\_MQCA  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_  
Renvoyé: Toujours

### ***CreateDate***

Description : Date de création de la file d'attente  
Identificateur : DATE DE CREATION\_MQCA  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_DATE\_MQ  
Renvoyé: Toujours

### ***CreateTime***

Description : Heure de création de la file d'attente  
Identificateur : HEURE DE CREATION\_MQCA  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours

### ***QType***

Description : Type de la file d'attente  
Identificateur : MQIA\_Q\_TYPE  
Type de données : MQCFIN  
Valeur : MQOT\_LOCAL  
Renvoyé: Toujours

### ***QDefinitionType***

Description : Type de définition de file d'attente  
Identificateur : MQIA\_DEFINITION\_TYPE  
Type de données : MQCFIN  
Valeurs : Les valeurs possibles sont les suivantes:

- MQQDT\_PRÉDÉFINI
- MQQDT\_PERMANENT\_DYNAMIQUE
- MQQDT\_TEMPORARY\_DYNAMIQUE

Renvoyé: Lorsqu'il est disponible

### ***QMinDepth***

Description :	Nombre minimal de lignes de la file d'attente pendant la période de surveillance
Identificateur :	MQIAMO_Q_MIN_DEPTH
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***QMaxDepth***

Description :	Longueur maximale de la file d'attente pendant la période de surveillance
Identificateur :	MQIAMO_Q_MAX_PROFONDEUR
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***AvgTimeOnQ***

Description :	Temps d'attente moyen, en microsecondes, des messages extraits de façon destructive de la file d'attente pendant la période de surveillance. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO64_AVG_Q_TIME
Type de données :	MQCFIL64
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***PutCount***

Description :	Nombre de messages persistants et non persistants correctement insérés dans la file d'attente, à l'exception des demandes MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir <a href="#">Note de référence 2</a> .
Identificateur :	MQIAMO_INSERTIONS
Type de données :	MQCFIL
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***PutFailCount***

Description :	Nombre de tentatives d'insertion d'un message dans la file d'attente ayant échoué
Identificateur :	MQIAMO_PUTS_FAILED
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QStatisticsData</i>

Renvoyé: Lorsqu'il est disponible

### ***Put1Count***

Description : Nombre de messages persistants et non persistants correctement insérés dans la file d'attente à l'aide d'appels MQPUT1 . Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO\_PUT1S

Type de données : MQCFIL

Inclus dans le groupe PCF: *QStatisticsData*

Renvoyé: Lorsqu'il est disponible

### ***Put1FailCount***

Description : Nombre de tentatives d'insertion de message ayant échoué à l'aide d'appels MQPUT1

Identificateur : MQIAMO\_PUT1S\_FAILED

Type de données : MQCFIN

Inclus dans le groupe PCF: *QStatisticsData*

Renvoyé: Lorsqu'il est disponible

### ***PutBytes***

Description : Nombre d'octets écrits dans les demandes d'insertion dans la file d'attente

Identificateur : MQIAMO64\_PUT\_BYTES

Type de données : MQCFIL64

Inclus dans le groupe PCF: *QStatisticsData*

Renvoyé: Lorsqu'il est disponible

### ***GetCount***

Description : Nombre de demandes d'extraction destructives réussies pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).

Identificateur : MQIAMO\_GETS

Type de données : MQCFIL

Inclus dans le groupe PCF: *QStatisticsData*

Renvoyé: Lorsqu'il est disponible

### ***GetFailCount***

Description : Nombre de demandes d'extraction destructives ayant échoué

Identificateur : Echec de MQIAMO\_GETS\_FAILED

Type de données : MQCFIN

Inclus dans le groupe PCF: *QStatisticsData*  
Renvoyé: Lorsqu'il est disponible

### **GetBytes**

Description : Nombre d'octets lus dans les demandes d'insertion destructive pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO64\_GET\_BYTES  
Type de données : MQCFIL64  
Inclus dans le groupe PCF: *QStatisticsData*  
Renvoyé: Lorsqu'il est disponible

### **BrowseCount**

Description : Nombre de demandes d'extraction non destructives réussies pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO\_NAVIGATIONS  
Type de données : MQCFIL  
Inclus dans le groupe PCF: *QStatisticsData*  
Renvoyé: Lorsqu'il est disponible

### **BrowseFailCount**

Description : Nombre de demandes d'extraction non destructives ayant échoué  
Identificateur : Echec de MQIAMO\_BROWSES\_FAILED  
Type de données : MQCFIN  
Inclus dans le groupe PCF: *QStatisticsData*  
Renvoyé: Lorsqu'il est disponible

### **BrowseBytes**

Description : Nombre d'octets lus dans les demandes d'extraction non destructives pour les messages persistants et non persistants. Ce paramètre est une liste d'entiers indexée par valeur de persistance. Voir [Note de référence 2](#).  
Identificateur : MQIAMO64\_BROWSE\_BYTES  
Type de données : MQCFIL64  
Inclus dans le groupe PCF: *QStatisticsData*  
Renvoyé: Lorsqu'il est disponible

### ***NonQueuedMsgCount***

Description :	Nombre de messages qui ont ignoré la file d'attente et ont été transférés directement à une application en attente.  Le contournement d'une file d'attente ne peut se produire que dans certaines circonstances. Ce nombre représente le nombre de fois où IBM MQ a pu ignorer la file d'attente et non le nombre de fois où une application a attendu.
Identificateur :	MQIAMO_MSGS_NOT_QUEUED
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***ExpiredMsgCount***

Description :	Nombre de messages persistants et non persistants qui ont été supprimés parce qu'ils avaient expiré avant d'avoir pu être extraits.
Identificateur :	MQIAMO_MSGS_EXPIRED
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***PurgeCount***

Description :	Nombre de messages purgés.
Identificateur :	MQIAMO_MSGS_PURGED
Type de données :	MQCFIN
Inclus dans le groupe PCF:	<i>QStatisticsData</i>
Renvoyé:	Lorsqu'il est disponible

### ***Données de message de statistiques de canal***

Utilisez cette page pour afficher la structure d'un message de statistiques de canal

Nom du message :	Message de statistiques de canal.
Plateformes :	Tous, sauf IBM MQ for z/OS.
File d'attente système:	SYSTEM.ADMIN.STATISTICS.QUEUE.

### ***QueueManager***

Description :	Nom du gestionnaire de files d'attente.
Identificateur :	MQCA_Q_MGR_NAME.
Type de données :	MQCFST.
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH.
Renvoyé:	Toujours.

### ***IntervalStartDate***

Description : Date de début de la période de surveillance.  
Identificateur : MQCAMO\_START\_DATE.  
Type de données : MQCFST.  
Longueur maximale : MQ\_DATE\_LENGTH.  
Renvoyé: Toujours.

### ***IntervalStartTime***

Description : Heure de début de la période de surveillance.  
Identificateur : MQCAMO\_START\_TIME.  
Type de données : MQCFST.  
Longueur maximale : MQ\_TIME\_LENGTH.  
Renvoyé: Toujours.

### ***IntervalEndDate***

Description : Date à la fin de la période de surveillance  
Identificateur : MQCAMO\_END\_DATE:  
Type de données : MQCFST.  
Longueur maximale : MQ\_DATE\_LENGTH.  
Renvoyé: Toujours.

### ***IntervalEndTime***

Description : Heure à la fin de la période de surveillance  
Identificateur : MQCAMO\_END\_TIME.  
Type de données : MQCFST.  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE  
Renvoyé: Toujours.

### ***CommandLevel***

Description : Niveau de commande du gestionnaire de files d'attente.  
Identificateur : MQIA\_COMMAND\_LEVEL.  
Type de données : MQCFIN.  
Renvoyé: Toujours.

### ***ObjectCount***

Description : Nombre d'objets canal consultés dans l'intervalle pour lequel des données statistiques ont été enregistrées. Cette valeur est définie sur le nombre de groupes PCF de données ChlStatisticscontenus dans le message.  
Identificateur : NOMBRE\_OBJETS\_MQIAMO\_OBJET

Type de données : MQCFIN.

Renvoyé: Toujours.

### ***ChlStatisticsData***

Description : Paramètres groupés spécifiant les détails des statistiques d'un canal.

Identificateur : MQGACF\_CHL\_STATISTICS\_DATA.

Type de données : MQCFGR.

Paramètres du groupe:

- ChannelName*
- ChannelType*
- RemoteQmgr*
- ConnectionName*
- MsgCount*
- TotalBytes*
- NetTimeMin*
- NetTimeAvg*
- NetTimeMax*
- ExitTimeMin*
- ExitTimeAvg*
- ExitTimeMax*
- FullBatchCount*
- IncplBatchCount*
- AverageBatchSize*
- PutRetryCount*

Renvoyé: Toujours.

### ***ChannelName***

Description : Nom du canal.

Identificateur : MQCACH\_CHANNEL\_NAME.

Type de données : MQCFST.

Longueur maximale : MQ\_CHANNEL\_NAME\_LENGTH.

Renvoyé: Toujours.

### ***ChannelType***

Description : Type de canal.

Identificateur : MQIACH\_CHANNEL\_TYPE.

Type de données : MQCFIN.



Valeurs : Les valeurs possibles sont les suivantes :

**EXPÉDITEUR\_MQCH**

Canal émetteur.

**SERVEUR\_MQ**

Canal du serveur.

**MQCHT\_RECEIVER**

Canal récepteur.

**DEMANDE\_MQCHT\_DEMANDEUR**

Canal demandeur

**MQCHT\_CLUSRCVR**

Canal récepteur de cluster.

**MQCHT\_CLUSSDR**

Canal émetteur de cluster.

Renvoyé: Toujours.

***RemoteQmgr***

Description : Nom du gestionnaire de files d'attente éloignées.

Identificateur : MQCA\_REMOTE\_Q\_MGR\_NAME.

Type de données : MQCFST.

Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

Renvoyé: Lorsqu'il est disponible.

***ConnectionName***

Description : Nom de connexion du gestionnaire de files d'attente éloigné.

Identificateur : MQCACH\_CONNECTION\_NAME.

Type de données : MQCFST

Longueur maximale : LONGUEUR\_NOM\_CONN\_MQ\_

Renvoyé: Lorsqu'il est disponible.

***MsgCount***

Description : Nombre de messages persistants et non persistants envoyés ou reçus.

Identificateur : MQIAMO\_MSGS.

Type de données : MQCFIN

Renvoyé: Lorsqu'il est disponible.

***TotalBytes***

Description : Nombre d'octets envoyés ou reçus pour les messages persistants et non persistants.

Identificateur : MQIAMO64\_BYTES.

Type de données : MQCFIN64.

Renvoyé: Lorsqu'il est disponible.

### ***NetTimeMin***

Description : Trajet aller-retour de canal enregistré le plus court, mesuré dans l'intervalle d'enregistrement, en microsecondes.

Identificateur : MQIAMO\_NET\_TIME\_MIN.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***NetTimeAvg***

Description : Durée moyenne de la boucle de canal enregistrée, mesurée dans l'intervalle d'enregistrement, en microsecondes.

Identificateur : MQIAMO\_NET\_TIME\_AVG.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***NetTimeMax***

Description : Boucle de canal enregistrée la plus longue, mesurée dans l'intervalle d'enregistrement, en microsecondes.

Identificateur : MQIAMO\_NET\_TIME\_MAX.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***ExitTimeMin***

Description : Temps enregistré le plus court, en microsecondes, consacré à l'exécution d'un exit utilisateur dans l'intervalle d'enregistrement,

Identificateur : MQIAMO\_EXIT\_TIME\_MIN.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***ExitTimeAvg***

Description : Temps moyen enregistré, en microsecondes, consacré à l'exécution d'un exit utilisateur au cours de l'intervalle d'enregistrement. Mesurée en microsecondes.

Identificateur : MQIAMO\_EXIT\_TIME\_AVG.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### ***ExitTimeMax***

Description : Temps enregistré le plus long, en microsecondes, consacré à l'exécution d'un exit utilisateur dans l'intervalle d'enregistrement. Mesurée en microsecondes.

Identificateur : MQIAMO\_EXIT\_TIME\_MAX.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### **FullBatchCount**

Description : Nombre de lots traités par le canal qui ont été envoyés car la valeur des attributs de canal BATCHSZ ou BATCHLIM a été atteinte.

Identificateur : MQIAMO\_FULL\_LOTS.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### **IncplBatchCount**

Description : Nombre de lots traités par le canal qui ont été envoyés sans que la valeur des attributs de canal BATCHSZ ou BATCHLIM soit atteinte.

Identificateur : MQIAMO\_INCOMPLETE\_LOTS.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### **AverageBatchSize**

Description : Taille moyenne des lots traités par le canal.

Identificateur : MQIAMO\_AVG\_BATCH\_SIZE.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### **PutRetryCount**

Description : Nombre de fois, dans l'intervalle de temps, qu'un message n'a pas pu être inséré et qu'il a été inséré dans une boucle de relance.

Identificateur : MQIAMO\_PUT\_RETRIES.

Type de données : MQCFIN.

Renvoyé: Lorsqu'il est disponible.

### **Remarques de référence**

Cette page permet d'afficher les remarques auxquelles les descriptions de la structure des messages de comptabilité et de statistiques font référence.

Les descriptions de données de message suivantes font référence à ces remarques:

- [«Données de message de comptabilité MQI», à la page 165](#)
- [«Données de message de comptabilité de file d'attente», à la page 177](#)
- [«Données de message de statistiques MQI», à la page 188](#)
- [«Données de message de statistiques de file d'attente», à la page 200](#)
- [«Données de message de statistiques de canal», à la page 206](#)

1. Ce paramètre est lié aux objets IBM MQ . Ce paramètre est un tableau de valeurs (MQCFIL ou MQCFIL64) indexé par les constantes suivantes:

<i>Tableau 24. Tableau indexé par type d'objet</i>	
Type d'objet	Contexte de valeur
MQOT_Q (1)	Contient la valeur relative aux objets file d'attente.

<i>Tableau 24. Tableau indexé par type d'objet (suite)</i>	
Type d'objet	Contexte de valeur
MQOT_NAMELIST (2)	Contient la valeur relative aux objets de liste de noms.
MQOT_PROCESS (3)	Contient la valeur relative aux objets de processus.
MQOT_Q_MGR (5)	Contient la valeur relative aux objets de gestionnaire de files d'attente.
MQOT_CHANNEL (6)	Contient la valeur relative aux objets de canal.
MQOT_AUTH_INFO (7)	Contient la valeur relative aux objets d'informations d'authentification.
MQOT_TOPIC (8)	Contient la valeur relative aux objets de rubrique.

**Remarque :** Un tableau de 13 valeurs MQCFIL ou MQCFIL64 est renvoyé, mais seules les valeurs répertoriées sont significatives.

2. Ce paramètre est lié aux messages IBM MQ . Ce paramètre est un tableau de valeurs (MQCFIL ou MQCFIL64) indexé par les constantes suivantes:

<i>Tableau 25. Tableau indexé par valeur de persistance</i>	
Constante	Valeur
1	Contient la valeur des messages non persistants.
2	Contient la valeur des messages persistants.

**Remarque :** L'index de chacun de ces tableaux commence à zéro, donc un index de 1 fait référence à la deuxième ligne du tableau. Les éléments de ces tableaux non répertoriés dans ces tableaux ne contiennent aucune information de comptabilité ou de statistiques.

## Trace de l'activité d'application

La trace de l'activité des applications produit des informations détaillées sur le comportement des applications connectées à un gestionnaire de files d'attente. Il trace le comportement d'une application et fournit une vue détaillée des paramètres utilisés par une application lors de son interaction avec les ressources IBM MQ . Elle affiche également la séquence des appels MQI émis par une application.


Utilisez la trace d'activité d'application lorsque vous avez besoin de plus d'informations que celles fournies par la surveillance des événements, la surveillance des messages, les messages de comptabilité et de statistiques et la surveillance en temps réel.

**Remarque :** La trace d'activité est générée sur la connexion IBM MQ pour chaque application ; par conséquent, si les messages de comptabilité sont activés, les opérations de trace d'activité sont comptabilisées dans les informations de comptabilité MQI pour chaque application.

IBM MQ prend en charge deux méthodes de collecte des données de trace d'activité d'application.

- Collecte centralisée des informations de trace d'activité d'application, où la trace d'activité d'application est collectée et lue en écrivant des messages PCF de trace d'activité dans la file d'attente système SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.
- Abonnement aux données de trace d'activité, écrit dans des rubriques système IBM MQ spéciales.

Notez que la trace d'activité n'est pas prise en charge par IBM MQ for z/OS.

 Comme vous pouvez désormais spécifier des noms d'application sur la plupart des langages de programmation pris en charge par IBM MQ, voir [Spécification du nom d'application dans les langages de programmation pris en charge](#) pour plus d'informations.

## Configuration de la collecte centralisée des informations de trace d'activité d'application

Un message de trace d'activité d'application est un message PCF. Vous pouvez configurer la trace d'activité à l'aide d'un fichier de configuration. Pour configurer la collecte centralisée des informations de trace d'activité d'application, vous devez définir l'attribut de gestionnaire de files d'attente ACTVTRC. Vous pouvez remplacer ce paramètre au niveau de la connexion à l'aide des options MQCONN ou au niveau de la strophe d'application à l'aide du fichier de configuration de trace d'activité.

### Pourquoi et quand exécuter cette tâche

Les messages de trace d'activité sont composés d'une structure MQMD: une structure d'en-tête PCF (MQCFH), suivie d'un certain nombre de paramètres PCF. Une séquence de groupes PCF de données ApplicationTracesuit les paramètres PCF. Ces groupes PCF collectent des informations sur les opérations MQI effectuées par une application lorsqu'elle est connectée à un gestionnaire de files d'attente. Vous configurez la trace de l'activité à l'aide d'un fichier de configuration appelé `mqt.ini`.

Pour contrôler la collecte ou non des informations de trace d'activité d'application, configurez un ou plusieurs des paramètres suivants:

1. Attribut du gestionnaire de files d'attente ACTVTRC.
2. Les paramètres ACTVCONO (dans la structure MQCNO transmise dans MQCONN).
3. Section correspondante de l'application dans le fichier de configuration de trace d'activité `mqt.ini`.

La séquence précédente est significative. L'attribut ACTVTRC est remplacé par les paramètres ACTVCONO, qui sont remplacés par les paramètres du fichier `mqt.ini`.

Les entrées de trace sont écrites après chaque opération, sauf indication contraire. Ces entrées sont d'abord écrites dans la file d'attente système SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE, puis écrit dans les messages de trace d'activité d'application lorsque l'application se déconnecte du gestionnaire de files d'attente. Pour les applications à exécution longue, des messages intermédiaires sont écrits si l'un des événements suivants se produit:

- La durée de vie de la connexion atteint une valeur de délai d'attente définie.
- Le nombre d'opérations atteint un nombre spécifié.
- La quantité de données collectées en mémoire atteint la longueur de message maximale autorisée pour la file d'attente.

Vous définissez la valeur de délai d'attente à l'aide du paramètre **ActivityInterval**. Vous définissez le nombre d'opérations à l'aide du paramètre **ActivityCount**. Les deux paramètres sont spécifiés dans le fichier de configuration de trace d'activité `mqt.ini`.

L'activation de la trace de l'activité de l'application peut affecter les performances. Le temps système peut être réduit en optimisant les paramètres **ActivityCount** et **ActivityInterval**. Voir [«Optimisation de l'impact sur les performances de la trace de l'activité de l'application»](#), à la page 221.

Le moyen le plus simple d'afficher le contenu des messages de trace d'activité d'application consiste à utiliser [«Exemple de programme amqsact»](#), à la page 222.

### Procédure

1. [«Définition de ACTVTRC pour contrôler la collecte des informations de trace d'activité»](#), à la page 214.
2. [«Définition des options MQCONN pour contrôler la collecte des informations de trace d'activité»](#), à la page 214.
3. [«Configuration du comportement de trace d'activité à l'aide de mqt.ini»](#), à la page 215.
4. [«Optimisation de l'impact sur les performances de la trace de l'activité de l'application»](#), à la page 221.

## Définition de **ACTVTRC** pour contrôler la collecte des informations de trace d'activité

Utilisez l'attribut de gestionnaire de files d'attente **ACTVTRC** pour contrôler la collecte des informations de trace d'activité d'application MQI

### Pourquoi et quand exécuter cette tâche

Les messages de trace d'activité d'application sont générés uniquement pour les connexions qui commencent après l'activation de la trace d'activité d'application. Le paramètre **ACTVTRC** peut avoir les valeurs suivantes:

#### Oui

La collecte de trace d'activité d'API est activée

#### Non

La collecte de trace d'activité d'API est désactivée

**Remarque :** Le paramètre **ACTVTRC** peut être remplacé par le paramètre **ACTVCONO** du gestionnaire de files d'attente. Si vous définissez le paramètre **ACTVCONO** sur **ENABLED**, le paramètre **ACTVTRC** peut être remplacé pour une connexion donnée à l'aide de la zone **Options** de la structure **MQCNO**. Voir [«Définition des options MQCONNX pour contrôler la collecte des informations de trace d'activité»](#), à la page 214.

### Exemple

Pour modifier la valeur du paramètre **ACTVTRC**, utilisez la commande **MQSC ALTER QMGR**. Par exemple, pour activer la collecte des informations de trace d'activité d'application MQI, utilisez la commande **MQSC** suivante:

```
ALTER QMGR ACTVTRC(ON)
```

### Que faire ensuite

Le moyen le plus simple d'afficher le contenu des messages de trace d'activité d'application consiste à utiliser [«Exemple de programme amqsact»](#), à la page 222.

L'activation de la trace de l'activité de l'application peut affecter les performances. Le temps système peut être réduit en optimisant les paramètres **ActivityCount** et **ActivityInterval**. Voir [«Optimisation de l'impact sur les performances de la trace de l'activité de l'application»](#), à la page 221.

## Définition des options **MQCONNX** pour contrôler la collecte des informations de trace d'activité

Si l'attribut de gestionnaire de files d'attente **ACTVCONO** est défini sur **ENABLED**, vous pouvez utiliser le paramètre **ConnectOpts** dans l'appel **MQCONNX** pour activer ou désactiver les rapports d'activité d'application par connexion. Ces options remplacent le comportement de trace d'activité défini par l'attribut de gestionnaire de files d'attente **ACTVTRC** et peuvent être remplacées par des paramètres du fichier de configuration de trace d'activité **mqt.ini**.

### Procédure

1. Définissez l'attribut de gestionnaire de files d'attente **ACTVCONO** sur **ENABLED**.

**Remarque :** Si une application tente de modifier le comportement comptable d'une application à l'aide du paramètre **ConnectOpts** et que l'attribut **QMGR ACTVCONO** est défini sur **DISABLED**, aucune erreur n'est renvoyée à l'application et la collecte de trace d'activité est définie par les attributs du gestionnaire de files d'attente ou le fichier de configuration de trace d'activité **mqt.ini**.

2. Définissez le paramètre **ConnectOpts** de l'appel **MQCONNX** sur **MQCNO\_ACTIVITY\_TRACE\_ENABLED**.

Le paramètre **ConnectOpts** de l'appel **MQCONNX** peut avoir les valeurs suivantes:

## **MQCNO\_ACTIVITY\_TRACE\_DISABLED**

La trace d'activité est désactivée pour la connexion.

## **MQCNO\_ACTIVITY\_TRACE\_ENABLED**

La trace d'activité est activée pour la connexion.

**Remarque :** Si une application sélectionne à la fois `MQCNO_ACTIVITY_TRACE_ENABLED` et `MQCNO_ACTIVITY_TRACE_DISABLED` pour `MQCONN`, l'appel échoue avec le code anomalie `MQRC_OPTIONS_ERROR`.

3. Vérifiez que ces paramètres de trace d'activité ne sont pas remplacés par les paramètres du fichier de configuration de trace d'activité `mqt.ini`.

Voir «[Configuration du comportement de trace d'activité à l'aide de `mqt.ini`](#)», à la page 215.

## **Que faire ensuite**

Le moyen le plus simple d'afficher le contenu des messages de trace d'activité d'application consiste à utiliser «[Exemple de programme `amqsact`](#)», à la page 222.

L'activation de la trace de l'activité de l'application peut affecter les performances. Le temps système peut être réduit en optimisant les paramètres **ActivityCount** et **ActivityInterval**. Voir «[Optimisation de l'impact sur les performances de la trace de l'activité de l'application](#)», à la page 221.

## **Configuration du comportement de trace d'activité à l'aide de `mqt.ini`**

Le comportement de la trace d'activité est configuré à l'aide d'un fichier de configuration appelé `mqt.ini`. Ce fichier est utilisé pour définir le niveau et la fréquence de génération de rapports sur les données de trace d'activité. Le fichier permet également de définir des règles pour activer et désactiver la trace d'activité en fonction du nom d'une application.

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

**Linux** **AIX** Sur les systèmes AIX and Linux, `mqt.ini` se trouve dans le répertoire de données du gestionnaire de files d'attente, qui est le même emplacement que le fichier `qm.ini`.

**Windows** Sur les systèmes Windows, `mqt.ini` se trouve dans le répertoire de données du gestionnaire de files d'attente `C:\Program Files\IBM\WebSphere MQ\mqmrs\queue_manager_name`. Les utilisateurs exécutant des applications à tracer doivent disposer des droits permettant de lire ce fichier.

**Remarque :** Le fichier `mqt.ini` des gestionnaires de files d'attente migrés à partir de IBM WebSphere MQ 7.1 ou d'une version antérieure sera manquant. Dans ce cas, le fichier `mqt.ini` doit être créé manuellement et les droits 660 doivent être définis sur le fichier.

Lorsque le fichier `mqt.ini` est modifié, les connexions IBM MQ nouvellement créées sont traitées en fonction de la version modifiée. Les connexions existantes continueront d'utiliser la version précédente sauf si les paramètres du gestionnaire de files d'attente sont modifiés, par exemple à la suite d'une commande `ALTER QMGR`.

Ce fichier suit le même format de paire de clé de section et de valeur de paramètre que les fichiers `mqs.ini` et `qm.ini`.

Le fichier se compose d'une seule section, **AllActivityTrace**, pour configurer le niveau et la fréquence des données de trace d'activité de rapport par défaut pour toutes les traces d'activité.

Le fichier peut également contenir plusieurs sections **ApplicationTrace**. Chacune d'elles définit une règle pour le comportement de trace d'une ou de plusieurs connexions, en fonction du nom d'application des connexions à la règle.

### **StropheAllActivityTrace**

Une seule section **AllActivityTrace** définit les paramètres de la trace d'activité qui est appliquée à toutes les connexions IBM MQ, sauf si elle est remplacée.

Les valeurs individuelles de la section **AllActivityTrace** peuvent être remplacées par des informations plus spécifiques dans une section **ApplicationTrace**.

Si plusieurs sections **AllActivityTrace** sont spécifiées, les valeurs de la dernière section sont utilisées. Les paramètres manquants dans la **AllActivityTrace** choisie prennent les valeurs par défaut. Les paramètres et les valeurs des sections **AllActivityTrace** précédentes sont ignorés.

Les paramètres suivants peuvent être spécifiés dans la section **AllActivityTrace** :

<i>Tableau 26. Paires paramètre / valeur pouvant être utilisées dans le fichier de configuration de trace d'activité</i>		
<b>Nom</b>	<b>Valeurs (valeur par défaut en gras)</b>	<b>Description</b>
ActivityInterval	0 à 99999999 ( <b>1</b> )	Intervalle de temps approximatif en secondes entre les messages de trace. Toutes les activités effectuées par une connexion dans cet intervalle seront écrites dans un seul message. Si la valeur est 0, le message trace est écrit lorsque la connexion est arrêtée (ou lorsque le nombre d'activités est atteint).
ActivityCount	0 à 99999999 ( <b>100</b> )	Nombre d'opérations MQI ou XA entre les messages de trace. Si cette valeur est 0, le message de trace est écrit lorsque la connexion se déconnecte (ou lorsque l'intervalle d'activité s'est écoulé).
TraceLevel	LOW / <b>MEDIUM</b> / HIGH	Quantité de détails de paramètre tracés pour chaque opération. La description des opérations individuelles indique les paramètres qui sont inclus pour chaque niveau de trace.
TraceMessageData	<b>0</b> -104 857 600 (maximum 100 Mo)	Quantité de données de message tracées en octets pour les opérations MQGET, MQPUT, MQPUT1 et de rappel
StopOnGetTraceMsg	<b>ON</b> / OFF	L'utilisation de la trace d'activité pour tracer les applications qui traitent également les messages de trace d'activité n'est pas recommandée en raison d'une éventuelle boucle.
SubscriptionDelivery	<b>BATCHED</b> / IMMEDIATE	Détermine si les paramètres ActivityInterval et ActivityCount doivent être utilisés lorsqu'un ou plusieurs abonnements à la trace de l'activité sont présents. La définition de ce paramètre sur IMMEDIATE entraîne le remplacement des valeurs ActivityInterval et ActivityCount par les valeurs effectives 1 lorsque les données de trace ont un abonnement correspondant. Les enregistrements de trace de l'activité ne sont pas regroupés avec d'autres enregistrements de la même connexion, mais sont distribués à l'abonnement immédiatement.

#### **sectionApplicationTrace**



Une section ApplicationTrace contient une règle qui définit les connexions IBM MQ qui seront tracées ou qui ne seront pas tracées en fonction du nom de l'application. Eventuellement, le comportement par défaut défini sous Allsettings qui remplace les paramètres de niveau de trace et de fréquence globaux.

Cette section peut inclure des paramètres ApplName, ApplFunction et ApplClass qui sont utilisés conformément aux règles de correspondance définies dans les règles de correspondance de connexion pour déterminer si la section s'applique à une connexion particulière ou non.

La section doit inclure le paramètre Trace pour déterminer si cette règle active ou désactive la trace d'activité pour les connexions correspondantes.

Une règle de désactivation peut être utilisée pour désactiver explicitement la trace pour des noms d'application plus spécifiques et pour remplacer le paramètre ACTVTRC des options de connexion du gestionnaire de files d'attente ou de la trace d'activité.

Les paramètres suivants peuvent être spécifiés dans la section **ApplicationTrace** :




<i>Tableau 27. Paires paramètre / valeur pouvant être utilisées dans le fichier de configuration de trace d'application</i>		
<b>Nom</b>	<b>Valeurs (valeur par défaut en gras)</b>	<b>Description</b>
Fonction de trace	ON / OFF (Paramètre requis-aucune valeur par défaut)	Commutateur de trace d'activité. Ce commutateur peut être utilisé dans la section spécifique à l'application pour déterminer si la trace d'activité est active pour la portée de la section d'application en cours. Notez que cette valeur remplace les paramètres ACTVTRC et ACTVCONO pour le gestionnaire de files d'attente.
ApplName	Chaîne de caractères (paramètre obligatoire-pas de valeur par défaut)	<p>Cette valeur est utilisée pour déterminer les applications auxquelles la strophe ApplicationTrace s'applique. Il est mis en correspondance avec la valeur ApplName de la structure de contexte d'exit d'API (qui est équivalente à MQMD.PutApplName). Le contenu de la valeur ApplName varie en fonction de l'environnement d'application.</p> <p>Pour les plateformes autres que z/OS, uniquement la partie nom de fichier de MQAXC.ApplName est mis en correspondance avec la valeur de la strophe. Les caractères situés à gauche du séparateur de chemin le plus à droite sont ignorés lors de la comparaison.</p> <p> Pour les applications z/OS, l'intégralité de MQAXC.ApplName est mis en correspondance avec la valeur de la strophe.</p> <p>Un seul caractère générique (*) peut être utilisé à la fin de la valeur ApplName pour correspondre à n'importe quel nombre de caractères après ce point. Si la valeur ApplName est définie sur un caractère générique unique (*), la valeur ApplName correspond à toutes les applications.</p>

Tableau 27. Paires paramètre / valeur pouvant être utilisées dans le fichier de configuration de trace d'application (suite)

Nom	Valeurs (valeur par défaut en gras)	Description
  ApplFunction	Chaîne de caractères (valeur par défaut <b>*</b> )	Cette valeur est utilisée pour déterminer les programmes d'application auxquels la section <b>ApplicationTrace</b> et la valeur <b>App1Name</b> s'appliquent.  La strophe est facultative, mais elle n'est valide que pour les gestionnaires de files d'attente IBM i . Un seul caractère générique (*) peut être utilisé à la fin de la valeur <b>App1Name</b> pour correspondre à n'importe quel nombre de caractères.  Par exemple, une section <b>ApplicationTrace</b> spécifiant <b>App1Name = *</b> et <b>App1Function = AMQSPUTO</b> s'applique à tous les appels du programme AMQSPUTO à partir de n'importe quel travail.
App1Class	USER / MCA / <b>TOUS</b>	Classe de l'application. Consultez le tableau suivant pour savoir comment les valeurs <b>AppType</b> correspondent aux connexions IBM MQ .

Le tableau suivant montre comment les valeurs *AppClass* correspondent aux zones **APICallerType** et **APIEnvironment** dans la structure de contexte d'exit d'API de connexion.

Tableau 28. Les valeurs de classe d'application et leur correspondance avec les zones APICallerType et APIEnvironment

CLASSE d'application	Type d'appelant de l'API:	Environnement d'API:	Description
UTILISATEUR	MQXACT_EXTERNAL	MQXE_AUTRES	Seules les applications utilisateur sont tracées
MCA	(Toute valeur)	MQXE_MCA MQXE_MCA_CLNTCONN MQXE_MCA_SVRCONN	Clients et canaux (amqrmppa)
TOUT	(Toute valeur)	(Toute valeur)	Toutes les connexions sont tracées



**Avertissement :** Vous devez utiliser un **APPLCLASS MCA** pour les applications utilisateur client, car la classe *USER* ne correspond pas à ces dernières.

Par exemple, pour tracer le modèle d'application **amqspu~~tc~~** , vous pouvez utiliser le code suivant:

```

ApplicationTrace:
App1Class=MCA                                     # Application type
                                                # Values: (USER | MCA | INTERNAL | ALL)
                                                # Default: USER
App1Name=amqsputc      # Application name (may be wildcard)
                                                # (matched to app name without path)
                                                # Default: *
Trace=ON                                           # Activity trace switch for application
                                                # Values: ( ON | OFF )
                                                # Default: OFF
ActivityInterval=30                               # Time interval between trace messages
                                                # Values: 0-99999999 (0=off)
    
```

```

ActivityCount=1           # Default: 0
                          # Number of operations between trace msgs
                          # Values: 0-99999999 (0=off)
TraceLevel=MEDIUM       # Default: 0
                          # Amount of data traced for each operation
                          # Values: LOW | MEDIUM | HIGH
TraceMessageData=1000   # Default: MEDIUM
                          # Amount of message data traced
                          # Values: 0-100000000
                          # Default: 0

```

La valeur par défaut `mqat.ini` générée lors de la création d'un gestionnaire de files d'attente contient une règle unique permettant de désactiver explicitement la trace d'activité pour l'exemple de trace d'activité fourni, **amqsact**.

## Règles de correspondance de connexion

Le gestionnaire de files d'attente applique les règles suivantes pour déterminer les paramètres de section à utiliser pour une connexion.

1. Une valeur spécifiée dans la section **AllActivityTrace** est utilisée pour la connexion, sauf si la valeur apparaît également dans une section **ApplicationTrace** et que la section remplit les critères de correspondance pour la connexion décrits aux points 2, 3 et 4.
2. **App1Class** est comparé au type de la connexion IBM MQ. Si **App1Class** ne correspond pas au type de connexion, la strophe est ignorée pour cette connexion.
3. La valeur *App1Name* de la strophe est comparée à la partie nom de fichier de la zone **App1Name** de la structure de contexte d'exit API (MQAXC) pour la connexion.

La partie du nom de fichier est dérivée des caractères situés à droite du séparateur de chemin final (/ ou \). Si la strophe **App1Name** inclut un caractère générique (\*), seuls les caractères situés à gauche du caractère générique sont comparés au nombre équivalent de caractères du **App1Name** de la connexion.

Par exemple, si une valeur de strophe "FRE\*" est spécifiée, seuls les trois premiers caractères sont utilisés dans la comparaison, donc "path/FREEDOM" et "path\FREDDY" correspondent, mais pas "path/FRIEND". Si la valeur *App1Name* de la section ne correspond pas à la connexion **App1Name**, la section est ignorée pour cette connexion.

4. Si plusieurs strophes correspondent aux **App1Name** et **App1Class** de la connexion, la strophe avec le **App1Name** le plus spécifique est utilisée.

Le **App1Name** le plus spécifique est défini comme celui qui utilise le plus de caractères pour correspondre au **App1Name** de la connexion.

Par exemple, si le fichier `ini` contient une strophe avec **App1Name** = "FRE\*" et une autre strophe avec **App1Name** = "FREE\*", la strophe avec **App1Name** = "FREE\*" est choisie comme meilleure correspondance pour une connexion avec **App1Name** = "path/FREEDOM" car elle correspond à quatre caractères (alors que **App1Name** = "FRE\*" ne correspond qu'à trois caractères).

5. Si, après l'application des règles aux points 2, 3 et 4, plusieurs sections correspondent aux connexions **App1Name** et **App1Class** de la connexion, les valeurs de la dernière correspondance sont utilisées et toutes les autres sections sont ignorées.

## Remplacement des paramètres par défaut pour chaque règle

Eventuellement, les paramètres de niveau et de fréquence de trace globale sous la section **AllActivityTrace** peuvent être remplacés pour les connexions correspondant à une section **ApplicationTrace**.

Les paramètres suivants peuvent être définis dans une section **ApplicationTrace**. Si elles ne sont pas définies, la valeur est héritée des paramètres de la section **AllActivityTrace** :

- **ActivityInterval**
- **ActivityCount**
- **TraceLevel**
- **TraceMessageData**

- **StopOnTraceMsg**

### Syntaxe mqat.ini

Les règles de syntaxe pour le format du fichier mqat.ini sont les suivantes:

- Le texte commençant par un hachage ou un point-virgule est considéré comme un commentaire qui s'étend jusqu'à la fin de la ligne.
- La première ligne significative (sans commentaire) doit être une clé de section.
- Une clé de strophe se compose du nom de la strophe, suivi d'un signe deux-points.
- Une paire paramètre-valeur se compose du nom d'un paramètre suivi d'un signe égal, puis de la valeur.
- Une seule paire paramètre-valeur peut apparaître sur une ligne. (Une valeur de paramètre ne doit pas être encapsulée sur une autre ligne).
- Les espaces de début et de fin sont ignorés. Il n'y a pas de limite sur la quantité d'espace entre les noms de section, les noms de paramètre et les valeurs, ou les paires paramètre / valeur. Les retours à la ligne sont significatifs et ne sont pas ignorés
- La longueur maximale d'une ligne est de 2048 caractères
- Les clés de section, les noms de paramètre et les valeurs de paramètre constantes ne sont pas sensibles à la casse, mais les valeurs de paramètre de variable (*ApplName* et *DebugPath*) sont sensibles à la casse.

### Exemple de fichier de trace d'activité d'application

L'exemple suivant montre comment les données de configuration sont spécifiées dans le fichier ini de trace d'activité.

```
AllActivityTrace:
ActivityInterval=1
ActivityCount=100
TraceLevel=MEDIUM
TraceMessageData=0
StopOnGetTraceMsg=ON

ApplicationTrace:
ApplName=amqs*
Trace=ON
TraceLevel=HIGH
TraceMessageData=1000

ApplicationTrace:
ApplName=amqsact*
Trace=OFF
```

La section **AllActivityTrace** ci-dessus définit le mode d'exécution par défaut de la trace d'activité lorsqu'elle est activée, via les règles ApplicationTrace ou l'attribut ACTVTRC du gestionnaire de files d'attente ou à l'aide d'un programme activé par une application.

La première section **ApplicationTrace** définit une règle qui génère une activité MQI par une application dont le nom commence par "amqs". La trace générée pour ces applications sera très détaillée et comprendra jusqu'à 1000 octets de données de message. L'intervalle d'activité et les paramètres de comptage seront hérités

La deuxième section **ApplicationTrace** définit une règle qui désactive la trace pour les applications dont les noms commencent par "amqsact" (exemple de trace d'activité). Cette règle remplace la règle on'précédente pour l'application amqsact, ce qui ne génère aucune trace pour cette application.

Un exemple est également fourni en tant qu'exemple appelé mqat.ini dans le répertoire des exemples C (le même répertoire que le fichier amqsact.c). Ce fichier peut être copié dans le répertoire de données du gestionnaire de files d'attente, pour les gestionnaires de files d'attente qui ont été migrés à partir d'une édition antérieure d'IBM MQ.

## Que faire ensuite

L'activation de la trace de l'activité de l'application peut affecter les performances. Le temps système peut être réduit en optimisant les paramètres **ActivityCount** et **ActivityInterval**. Voir [«Optimisation de l'impact sur les performances de la trace de l'activité de l'application»](#), à la page 221.

### ***Optimisation de l'impact sur les performances de la trace de l'activité de l'application***

L'activation de la trace d'activité d'application peut entraîner une baisse des performances. Cela peut être réduit en traçant uniquement les applications dont vous avez besoin, en augmentant le nombre d'applications qui drainent la file d'attente et en optimisant **ActivityInterval**, **ActivityCount** et **TraceLevel** dans `mqat.ini`.

## Pourquoi et quand exécuter cette tâche

L'activation sélective de la trace d'activité d'application pour une application ou pour toutes les applications de gestionnaire de files d'attente peut entraîner une activité de messagerie supplémentaire, et le gestionnaire de files d'attente nécessite un espace de stockage supplémentaire. Dans les environnements où les performances de messagerie sont critiques, par exemple dans les applications à charge de travail élevée ou où un accord sur les niveaux de service (SLA) requiert un temps de réponse minimal de la part du fournisseur de messagerie, il peut ne pas être approprié de collecter la trace de l'activité de l'application ou il peut être nécessaire d'ajuster le détail ou la fréquence des messages d'activité de trace générés. Les valeurs prédéfinies de **ActivityInterval**, **ActivityCount** et **TraceLevel** dans le fichier `mqat.ini` fournissent un équilibre par défaut entre les détails et les performances. Toutefois, vous pouvez ajuster ces valeurs pour répondre aux exigences fonctionnelles et de performances précises de votre système.

## Procédure

- Ne tracez que les applications dont vous avez besoin.

Pour ce faire, créez une strophe spécifique à l'application `ApplicationTrace` dans `mqat.ini` ou modifiez l'application pour spécifier `MQCNO_ACTIVITY_TRACE_ENABLED` dans la zone d'options de la structure **MQCNO** sur un appel `MQCONN`. Voir [«Configuration du comportement de trace d'activité à l'aide de `mqat.ini`»](#), à la page 215 et [«Définition des options `MQCONN` pour contrôler la collecte des informations de trace d'activité»](#), à la page 214.

- Avant de démarrer la trace, vérifiez qu'au moins une application est en cours d'exécution et qu'elle est prête à extraire les données de message de trace d'activité du `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`.
- Maintenez la longueur de la file d'attente aussi faible que possible, en augmentant le nombre d'applications drainant la file d'attente.
- Définissez la valeur **TraceLevel** dans le fichier `mqat.ini` pour collecter la quantité minimale de données requise.

`TraceLevel=LOW` a l'impact le plus faible sur les performances de messagerie. Voir [«Configuration du comportement de trace d'activité à l'aide de `mqat.ini`»](#), à la page 215.

- Optimisez les valeurs **ActivityCount** et **ActivityInterval** dans `mqat.ini` pour ajuster la fréquence de génération des messages de trace d'activité.

Si vous tracez plusieurs applications, les messages de trace d'activité risquent d'être produits plus rapidement qu'ils ne peuvent être supprimés du `SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE`. Toutefois, lorsque vous réduisez la fréquence de génération des messages trace d'activité, vous augmentez également l'espace de stockage requis par le gestionnaire de files d'attente et la taille des messages lorsqu'ils sont écrits dans la file d'attente.

## Que faire ensuite

Le moyen le plus simple d'afficher le contenu des messages de trace d'activité d'application consiste à utiliser [«Exemple de programme `amqsact`»](#), à la page 222.

## Exemple de programme amqsact

**amqsact** formate les messages Application Activity Trace pour vous et est fourni avec IBM MQ.

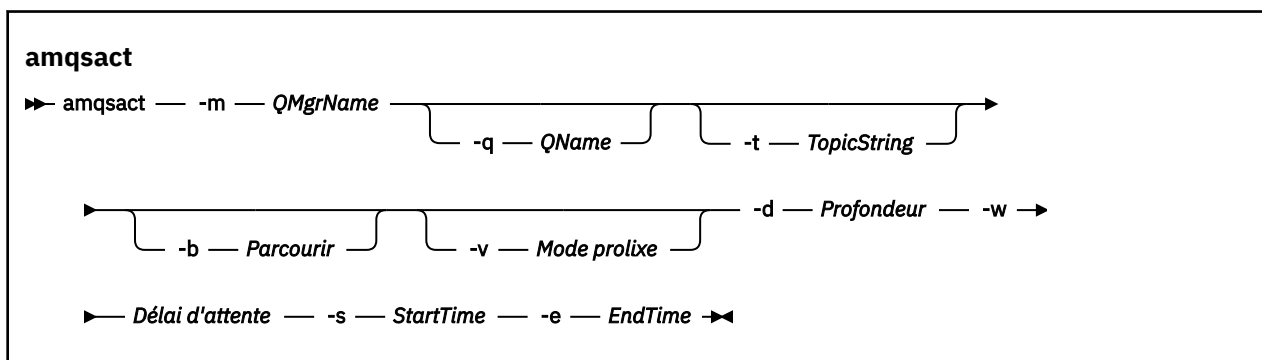
Le programme compilé se trouve dans le répertoire des exemples:

- **Linux** **AIX** Sous AIX and Linux MQ\_INSTALLATION\_PATH/samp/bin
- **Windows** Sous Windows MQ\_INSTALLATION\_PATH\tools\c\Samples\Bin

## Mode affichage

Par défaut, **amqsact** en mode affichage traite les messages sur SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Vous pouvez changer ce comportement en spécifiant un nom de file d'attente ou une chaîne de rubrique.

Vous pouvez également contrôler la période de trace affichée et indiquer si les messages de trace d'activité sont supprimés ou conservés après l'affichage.



## Paramètres requis pour le mode d'affichage

### -m *NomGestF*

Nom du gestionnaire de files d'attente.

### -d *profondeur*

Nombre d'enregistrements à afficher.

### -w *délai*

Délai d'attente, en secondes. Si aucun message de trace n'apparaît dans la période spécifiée, **amqsact** se ferme.

### -s *StartTime*

Heure de début de l'enregistrement à traiter.

### -e *EndTime*

Heure de fin de l'enregistrement à traiter.

## Paramètres facultatifs pour le mode d'affichage

### -q *nom\_qualifié*

Indiquez une file d'attente spécifique pour remplacer le nom de file d'attente par défaut

### -t *TopicString*

S'abonner à une rubrique d'événement

### -b

Parcourir uniquement les enregistrements

### -v

Sortie prolix

## Exemple de sortie pour le mode d'affichage

Utilisez **amqsact** sur le gestionnaire de files d'attente *TESTQM*, avec sortie prolix, sur un appel d'API MQCONN:

```
amqsact -m TESTQM -v
```

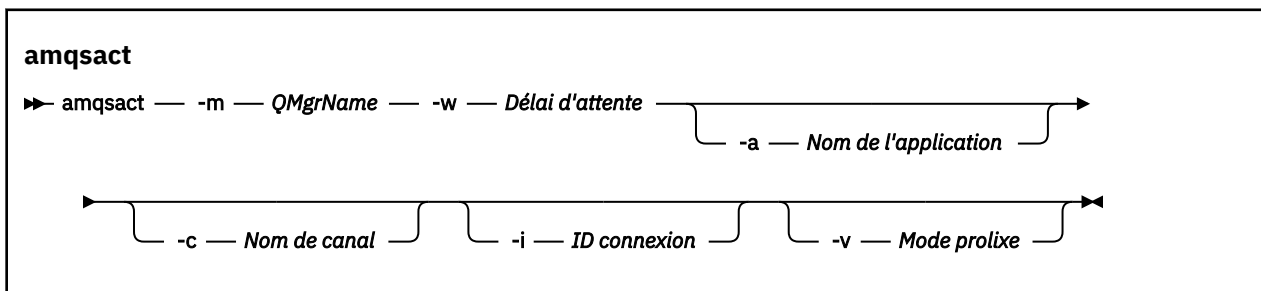
La commande précédente fournit l'exemple de sortie suivant:

```
MonitoringType: MQI Activity Trace
Correl_id:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM  '
00000010: B5F6 4251 2000 E601
QueueManager: 'TESTQM'
Host Name: 'ADMINIB-1VTJ6N1'
IntervalStartDate: '2014-03-15'
IntervalStartTime: '12:08:10'
IntervalEndDate: '2014-03-15'
IntervalEndTime: '12:08:10'
CommandLevel: 750
SeqNumber: 0
ApplicationName: 'IBM MQ_1\bin\amqsput.exe'
Application Type: MQAT_WINDOWS_7
ApplicationPid: 14076
UserId: 'Emma_Bushby'
API Caller Type: MQXACT_EXTERNAL
API Environment: MQXE_OTHER
Application Function: ''
Appl Function Type: MQFUN_TYPE_UNKNOWN
Trace Detail Level: 2
Trace Data Length: 0
Pointer size: 4
Platform: MQPL_WINDOWS_7
MQI Operation: 0
Operation Id: MQXF_CONN
ApplicationTid: 1
OperationDate: '2014-03-15'
OperationTime: '12:08:10'
ConnectionId:
00000000: 414D 5143 5445 5354 514D 2020 2020 2020 'AMQCTESTQM  '
00000010: FFFFFFFB5FFFFFFFFF6 4251 2000 FFFFFFFE601  '
QueueManager: 'TESTQM'
Completion Code: MQCC_OK
Reason Code: 0
```

## Mode dynamique

Vous pouvez activer le mode dynamique en spécifiant un nom d'application, un nom de canal ou un identificateur de connexion comme argument de **amqsact**. Notez que vous pouvez utiliser des caractères génériques dans le nom.

En mode dynamique, les données de trace d'activité sont activées au début de l'exemple à l'aide d'un abonnement non durable à une rubrique système. La collecte des données de trace d'activité s'arrête lorsqu'**amqsact** s'arrête. Vous devez spécifier un délai pour **amqsact** en mode dynamique. Vous pouvez exécuter plusieurs copies de **amqsact** simultanément, chaque instance recevant une copie des données de trace d'activité.



## Paramètres requis pour le mode dynamique

### -m *NomGestF*

Nom du gestionnaire de files d'attente.

### -w *délai*

Délai d'attente, en secondes. Si aucun message de trace n'apparaît dans la période spécifiée, **amqsact** se ferme.

## Paramètres facultatifs pour le mode dynamique

### -a *Nom de l'application*

Indiquez un nom d'application pour lequel collecter les messages

### -c *Nom de canal*

Indiquez un canal pour lequel collecter les messages

### -i *ID connexion*

Spécifiez la connexion pour laquelle collecter les messages.

### -v

Sortie prolixe

## Exemple de sortie pour le mode dynamique

La commande suivante génère et affiche des messages trace d'activité pour toutes les connexions établies par les applications qui commencent par le texte "amqs". Au bout de 30 secondes d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

La commande ci-après génère et affiche des messages trace d'activité pour toutes les activités sur le canal QMGR1.TO.QMGR2. Au bout de 10 secondes d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

La commande ci-après génère et affiche des messages trace d'activité prolixes pour toutes les activités sur la connexion IBM MQ existante dont l'objet CONN est "6B576B5420000701" et l'objet EXTCONN est "414D5143514D47523120202020202020". Au bout d'une minute d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D4752312020202020202020206B576B5420000701 -v
```

## Abonnement aux informations de trace d'activité d'application

Vous pouvez vous abonner dynamiquement aux informations de trace d'activité d'application comme alternative à la collecte des informations via la configuration au niveau du gestionnaire de files d'attente.

## Pourquoi et quand exécuter cette tâche

La trace de l'activité d'application effectue un suivi du comportement d'une application et fournit une vue détaillée des paramètres qui sont utilisés par une application lorsqu'elle interagit avec des ressources IBM MQ. Elle affiche également la séquence des appels MQI émis par une application.

En plus d'écrire des données de trace dans la file d'attente système, le produit permet de s'abonner dynamiquement aux données de trace d'activité, écrites dans des rubriques système IBM MQ spéciales, plutôt que de collecter les informations via la configuration au niveau du gestionnaire de files d'attente.



Notez que depuis IBM MQ 9.0 , le produit n'utilise pas d'exits à cette fin. Si vous avez déjà utilisé des exits pour tracer l'activité de l'application, vous devez utiliser la méthode de remplacement pour collecter la trace de l'activité de l'application.

La création d'un abonnement active la trace de l'activité. Il n'est pas nécessaire de définir des attributs de gestionnaire de files d'attente ou d'application, comme pour la collecte centrale des données de trace. Toutefois, tout blocage explicite de la trace de l'activité via la désactivation de la trace au niveau du gestionnaire de files d'attente ou de l'application empêche la distribution de la trace de l'activité aux abonnements correspondants.

## Procédure

- [«Abonnements à la trace de l'activité d'application»](#), à la page 225
- [«Création d'abonnements à une trace d'activité d'application»](#), à la page 225
- [«Utilisation d'amqsact pour afficher les messages trace»](#), à la page 227
- [«Configuration des niveaux de trace à l'aide de mqat.ini»](#), à la page 229

### ***Abonnements à la trace de l'activité d'application***

Vous pouvez vous abonner à une rubrique système IBM MQ afin de collecter des informations de trace de l'activité d'application.

Vous vous abonnez à une chaîne de rubrique système IBM MQ spéciale qui représente l'activité dont vous voulez effectuer le suivi. L'abonnement génère automatiquement des messages de données de trace de l'activité et les publie dans la file d'attente de destination de l'abonnement. Si vous supprimez l'abonnement, la génération des données de trace de l'activité s'arrête pour cet abonnement.

Un abonnement peut effectuer le suivi de l'activité sur l'une des ressources suivantes :

- Une application spécifiée
- Un canal IBM MQ spécifié
- Une connexion IBM MQ existante

Vous pouvez créer plusieurs abonnements, avec des chaînes de rubrique identiques ou différentes. Si vous créez plusieurs abonnements avec les mêmes chaînes de rubrique de trace de l'activité système, chaque abonnement reçoit une copie des données de trace de l'activité, ce qui peut avoir un impact négatif sur les performances.

L'activation d'un niveau de trace d'activité peut avoir un impact négatif sur les performances. Plus le nombre d'abonnements est élevé, ou plus le nombre de ressources auxquelles vous êtes abonné est élevé, plus les temps système potentiels sont élevés. Pour réduire les temps système de la collecte des données de trace de l'activité, les données sont écrites dans des messages et distribuées aux abonnements de façon asynchrone par rapport à l'activité d'application elle-même. Souvent, plusieurs opérations sont écrites dans un seul message de données de trace de l'activité. L'opération asynchrone peut générer un délai entre l'opération de l'application et la réception des données de trace enregistrant l'opération.

### ***Création d'abonnements à une trace d'activité d'application***

Vous pouvez créer des abonnements à des rubriques spécifiques pour collecter des données de trace d'activité d'application.

Lorsqu'un abonnement est créé pour des chaînes de rubrique système spécifiques, des messages de données de format de commande programmable (PCF) de trace d'activité appropriés sont publiés automatiquement dans cet abonnement. Pour plus d'informations sur l'abonnement aux rubriques, voir [Messagerie de publication / abonnement](#).

Le format des chaînes de rubrique est le suivant :

```
$SYS/MQ/INFO/QMGR/qmgr_name/ActivityTrace/resource_type/resource_identifieur
```

où :

- *nom\_gest\_files\_attente* spécifie le gestionnaire de files d'attente auquel l'application tracée est connectée. *qmgr\_name* est le nom du gestionnaire de files d'attente avec tous les caractères blancs de fin supprimés et toutes les barres obliques (/) remplacées par une perluète (&) caractère.
- *type\_ressource* spécifie le type de ressource pour lequel des données sont collectées et correspond à l'une des chaînes suivantes :
  - *AppName* pour spécifier une application. La demande s'abonne à toutes les connexions IBM MQ dont le nom d'application correspond à celui qui est spécifié par *identificateur\_ressource*.
  - *ChannelName* pour spécifier un canal IBM MQ .
  - *ConnectionId* pour spécifier une connexion IBM MQ .
- *identificateur\_ressource* identifie la ressource réelle. Le format dépend du type de ressource :
  - Pour un type de ressource *AppName*, *resource\_identifier* est la partie de fin (valeur qui suit le dernier / ou \) du nom d'application tel qu'il est vu par le gestionnaire de files d'attente, avec tous les caractères blancs de fin supprimés. La valeur correspond à la valeur de *nom\_app* de la structure de contexte d'exit API (MQAXC). La valeur *AppName* d'une connexion est renvoyée en tant que valeur *APPLTAG* lorsque vous utilisez la commande MQSC **DISPLAY CONN**.
  - Pour le type de ressource *nom\_canal*, *identificateur\_ressource* est le nom du canal à tracer. Si le nom de canal identifie un canal SVRCONN, l'intégralité de l'activité de l'application pour les clients connectés est tracée. Si le nom de canal identifie un gestionnaire de files d'attente sur un canal de gestionnaire de files d'attente, les messages entrants et sortants sont tracés. *resource\_identifier* est le nom du canal avec tous les caractères blancs de fin supprimés et tous les caractères '/' remplacés par un caractère '&' caractère.
  - Pour un type de ressource *ConnectionId*, *resource\_identifier* est l'identificateur de connexion unique affecté à chaque connexion. L'identificateur de connexion dans la chaîne de rubrique est la valeur de 24 octets complète écrite sous forme de chaîne hexadécimale. Il s'agit de la concaténation de *EXTCONN* suivi des valeurs *CONN* renvoyées par la commande MQSC **DISPLAY CONN**.

Vous pouvez utiliser des caractères génériques dans *identificateur\_ressource* pour faire correspondre plusieurs identités de ressource dans un abonnement unique. Le caractère générique peut être dans le style de rubrique par défaut ('#'ou'+') ou dans le style de caractère ('\*'ou'?'). Lorsque vous utilisez un caractère générique de style rubrique, vous ne pouvez pas le combiner avec une partie d'un nom de ressource ; il ne peut être utilisé que pour rechercher toutes les applications, les canaux ou les connexions possibles. L'utilisation de caractères génériques augmente le niveau des données de trace qui sont générées, ce qui peut avoir un impact sur les performances.

Pour vous abonner à ces chaînes de rubrique, vous devez disposer de l'autorisation "S'abonner". Les rubriques système n'héritent pas des autorisations de la racine de l'arborescence de rubriques du gestionnaire de files d'attente. Un utilisateur peut obtenir l'accès à un objet de rubrique administré au point *\$\$SYS/MQ* ou au-delà dans l'arborescence de rubriques. Vous pouvez vous abonner si vous avez accès à *SYSTEM.ADMIN.TOPIC*, bien que cette opération accorde l'accès à toutes les chaînes de rubrique *\$\$SYS/MQ*, et pas seulement à la trace d'activité. Pour contrôler l'accès plus précisément, vous pouvez définir de nouveaux objets de rubrique administrés pour des points plus profonds dans l'arborescence, pour l'intégralité de la trace d'activité ou par exemple pour un nom d'application ou un nom de canal spécifique.

## Exemples

L'exemple suivant illustre une chaîne de rubrique pour une application nommée *amqsput*, qui s'exécute sur un système Windows :

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqsputc.exe
```

L'exemple suivant illustre une chaîne de rubrique pour un canal :

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/SYSTEM.DEF.SVRCONN
```

L'exemple suivant illustre une chaîne de rubrique pour une connexion :

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ConnectionId/  
414D5143514D4752312020202020206B576B5420000701
```

L'exemple suivant illustre une chaîne de rubrique qui crée un abonnement afin de tracer les données pour tous les canaux sur le gestionnaire de files d'attente QMGR1 :

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/ChannelName/#
```

L'exemple suivant illustre une chaîne de rubrique qui crée un abonnement afin de tracer les données pour toutes les applications dont le nom commence par "amqs" (notez que pour que le caractère générique "\*" puisse être utilisé, l'abonnement doit être créé avec le modèle de caractère générique) :

```
$SYS/MQ/INFO/QMGR/QMGR1/ActivityTrace/AppName/amqs*
```

### Concepts associés

«Rubriques système pour la surveillance et la trace de l'activité», à la page 301

Les rubriques système dans les arborescences de rubriques du gestionnaire de files d'attente sont utilisées pour la surveillance des ressources (dont certaines sont similaires au contenu des messages de statistiques) et comme moyen de consommer la trace de l'activité de l'application.

### Utilisation d'amqsact pour afficher les messages trace

Vous pouvez utiliser le **amqsact** programme pour générer et afficher des messages de trace.

Le programme **amqsact** est un exemple d'IBM MQ. Pour utiliser cet exemple, vous devez utiliser le fichier exécutable connecté au client, **amqsactc**. Le fichier exécutable se trouve dans le répertoire des exemples :

- Sur les plateformes Linux et UNIX, `MQ_INSTALLATION_PATH/samp/bin64`
- Sur les plateformes Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`

Vous pouvez utiliser **amqsact** de deux façons :

#### Mode affichage

Formatez et affichez les messages de données de trace d'activité qui sont distribués à SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

#### Mode dynamique

Créez un abonnement à un ensemble de ressources et affichez la trace d'activité générée en exécutant **amqsact**.

### Mode affichage

Par défaut, **amqsact** en mode affichage traite les messages sur SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE. Vous pouvez changer ce comportement en spécifiant un nom de file d'attente ou une chaîne de rubrique. La trace d'activité doit être activée à l'aide de l'une des méthodes décrites dans [Collecte des informations de trace d'activité d'application](#). Vous pouvez contrôler la période de trace qui est affichée et spécifier si les messages trace d'activité doivent être supprimés ou conservés après leur affichage. En mode affichage, **amqsact** admet les arguments suivants :

#### -m *nom\_gest\_files\_attente*

Obligatoire. Spécifiez le gestionnaire de files d'attente pour lequel les messages trace sont collectés.

#### -q *nom\_file\_attente*

Affichez seulement les messages trace qui concernent la file d'attente nommée.

#### -t *chaîne\_rubrique*

Affichez seulement les messages trace qui concernent la rubrique nommée.

#### -b

Indiquez que les messages trace doivent être conservés après leur affichage.

**-v**

Affichez les messages trace en mode prolix.

**-d profondeur**

Nombre de messages à afficher.

**-w délai**

Spécifiez un délai d'attente. Si aucun message trace n'apparaît au cours de cette période, **amqsact** se ferme.

**-s heure\_début**

Utilisez cet argument avec l'argument **-e** pour spécifier une période. Les messages trace publiés au cours de cette période sont affichés.

**-e heure\_fin**

Utilisez cet argument avec l'argument **-s** pour spécifier une période. Les messages trace publiés au cours de cette période sont affichés.

Par exemple, la commande suivante affiche des messages trace d'activité qui sont conservés dans SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE et les supprime après leur affichage :

```
amqsact -m QMGR1
```

La commande ci-après affiche les messages trace d'activité de la file d'attente spécifiée, SUB.QUEUE, et les supprime après leur affichage. Les messages restent affichés jusqu'à écoulement d'une période de 30 secondes sans nouveau message. Par exemple, cette commande peut être utilisée avec un abonnement à une chaîne de rubrique système de trace d'activité.

```
amqsact -m QMGR1 -q SUB.QUEUE.1 -w 30
```

La commande ci-après affiche au format prolix toutes les données de trace d'activité conservées dans SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE et qui ont été générées au cours de la période de 20 minutes spécifiée. Les messages restent dans la file d'attente après leur affichage.

```
amqsact -m QMGR1 -b -v -s 2014-12-31 23.50.00 -e 2015-01-01 00.10.00
```

## Mode dynamique

Vous activez le mode dynamique en spécifiant un nom d'application, un nom de canal ou un identificateur de connexion comme argument d'**amqsact**. L'utilisation de caractères génériques est autorisée dans le nom. En mode dynamique, les données de trace d'activité sont activées au démarrage de l'exemple à l'aide d'un abonnement non durable à une rubrique système. La collecte des données de trace d'activité s'arrête lorsqu'**amqsact** s'arrête. Vous devez spécifier un délai pour **amqsact** en mode dynamique. Vous pouvez exécuter plusieurs copies d'**amqsact** simultanément, et chaque instance reçoit une copie des données de trace d'activité. En mode dynamique, **amqsact** admet les arguments suivants :

**-m nom\_gest\_files\_attente**

Obligatoire. Spécifiez le gestionnaire de files d'attente pour lequel les messages trace sont collectés.

**-w délai**

Obligatoire. Spécifiez un délai d'attente. Si aucun message trace n'apparaît au cours de cette période, **amqsact** se ferme.

**-a nom\_application**

Spécifiez l'application pour laquelle collecter les messages.

**-c nom\_canal**

Spécifiez le canal pour lequel collecter les messages.

**-i ID\_connexion**

Spécifiez la connexion pour laquelle collecter les messages.

**-v**

Affichez les messages trace en mode prolix.

Par exemple, la commande ci-après génère et affiche des messages trace d'activité pour toutes les connexions établies par des applications dont le nom est "amqsget.exe". Au bout de 30 secondes d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 30 -a amqsget.exe
```

La commande ci-après génère et affiche des messages trace d'activité pour toutes les connexions établies par des applications qui commencent par le texte "amqs". Au bout de 30 secondes d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 30 -a amqs*
```

La commande ci-après génère et affiche des messages trace d'activité pour toutes les activités sur le canal QMGR1.TO.QMGR2. Au bout de 10 secondes d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 10 -c QMGR1.TO.QMGR2
```

La commande ci-après génère et affiche des messages trace d'activité pour toutes les activités sur tous les canaux. Au bout de 10 secondes d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 10 -c #
```

La commande ci-après génère et affiche des messages trace d'activité prolixes pour toutes les activités sur la connexion IBM MQ existante dont l'objet CONN est "6B576B5420000701" et l'objet EXTCONN est "414D5143514D47523120202020202020". Au bout d'une minute d'inactivité, le programme **amqsact** s'arrête, et aucune nouvelle donnée de trace d'activité n'est générée.

```
amqsactc -m QMGR1 -w 60 -i 414D5143514D475231202020202020206B576B5420000701 -v
```

### ***Configuration des niveaux de trace à l'aide de mqat.ini***

Vous configurez les niveaux de trace pour un gestionnaire de files d'attente en définissant des valeurs pour la strophe AllActivityTrace du fichier de configuration mqat.ini.

Vous pouvez définir les valeurs suivantes pour la section AllActivityTrace :

#### **ActivityInterval**

Intervalle en secondes entre deux messages trace. La trace de l'activité n'utilise pas d'unité d'exécution de type minuteur ; par conséquent, le message trace n'est pas écrit dès que l'intervalle est écoulé, mais lorsque la première opération MQI est exécutée une fois l'intervalle de temps écoulé. Si la valeur est 0, le message trace est écrit lorsque la connexion est arrêtée (ou lorsque le nombre d'activités est atteint). La valeur par défaut est 1.

#### **ActivityCount**

Nombre d'opérations MQI entre les messages trace. Si la valeur est 0, le message trace est écrit lorsque la connexion est arrêtée (ou lorsque l'intervalle de l'activité est écoulé). La valeur par défaut est 100.

#### **TraceLevel**

Niveau de détail des paramètres pour le suivi de chaque opération. La description des opérations individuelles indique les paramètres qui sont inclus pour chaque niveau de trace. Choisissez LOW, MEDIUM ou HIGH. La valeur par défaut est MEDIUM.

### **TraceMessageData**

Quantité de données de message dont le suivi est assuré en octets pour les opérations MQGET, MQPUT, MQPUT1 et de rappel. La valeur par défaut est 0.

### **StopOnGetTraceMsg**

La valeur peut être ON ou OFF. Par défaut, il s'agit de ON.

### **SubscriptionDelivery**

La valeur peut être BATCHED ou IMMEDIATE. Détermine si les paramètres ActivityInterval et ActivityCount doivent être utilisés lorsqu'un ou plusieurs abonnements à la trace de l'activité sont présents. La définition de ce paramètre sur IMMEDIATE entraîne le remplacement des valeurs ActivityInterval et ActivityCount par les valeurs effectives 1 lorsque les données de trace ont un abonnement correspondant. Les enregistrements de trace de l'activité ne sont pas regroupés avec d'autres enregistrements de la même connexion, mais sont distribués à l'abonnement immédiatement. La valeur IMMEDIATE augmente les temps système de collecte des données de trace de l'activité. La valeur par défaut est BATCHED.

## **Référence de message de trace d'activité d'application**

Cette page permet d'obtenir une présentation du format des messages trace d'activité d'application et des informations renvoyées dans ces messages.

Les messages de trace d'activité d'application sont des messages IBM MQ standard contenant un descripteur de message et des données de message. Les données de message contiennent des informations sur les opérations MQI effectuées par les applications IBM MQ ou sur les activités qui se produisent dans un système IBM MQ .

### **Descripteur de message**

- Une structure MQMD

### **Données de message**

- En-tête PCF (MQCFH)
- Données de message de trace d'activité d'application toujours renvoyées
- Données de message de trace d'activité d'application spécifiques à l'opération

### **Message de trace d'activité d'application MQMD (descripteur de message)**

Cette page permet de comprendre les différences entre le descripteur de message des messages de trace d'activité d'application et le descripteur de message des messages d'événement.

Les paramètres et les valeurs du descripteur de message du message de trace d'activité d'application sont identiques à ceux du descripteur de message des messages d'événement, à l'exception suivante:

#### **Format**

Description : Nom de format des données de message.

Valeur : **MQFMT\_ADMIN**  
Message d'administration.

#### **CorrelId**

Description : Identificateur de corrélation.

Valeur : Initialisé avec l'ID ConnectionId de l'application

### **MQCFH (en-tête PCF)**

Cette page permet d'afficher les valeurs PCF contenues dans la structure MQCFH pour un message de trace d'activité

Pour un message de trace d'activité, la structure MQCFH contient les valeurs suivantes:

### **Type**

Description : Type de structure qui identifie le contenu du message.  
Type de données : MQLONG:  
Valeur : MQCFT\_APP\_ACTIVITE

### **StrucLength**

Description : Longueur en octets de la structure MQCFH.  
Type de données : MQLONG:  
Valeur : LONGUEUR\_STRUC\_MQCFH

### **Version**

Description : Numéro de version de la structure.  
Type de données : MQLONG:  
Valeurs : MQCFH\_VERSION\_3

### **Command**

Description : Identificateur de commande. Cette zone identifie la catégorie du message.  
Type de données : MQLONG:  
Valeurs : MQCMD\_ACTIVITY\_TRACE

### **MsgSeqNumber**

Description : Numéro de séquence du message. Cette zone indique le numéro de séquence du message dans un groupe de messages associés.  
Type de données : MQLONG:  
Valeurs : 1

### **Control**

Description : Options de contrôle.  
Type de données : MQLONG:  
Valeurs : MQCFC\_LAST.

### **CompCode**

Description : Code achèvement.  
Type de données : MQLONG:  
Valeurs : MQCC\_OK.

### **Reason**

Description : Code raison qualifiant le code achèvement.  
Type de données : MQLONG:  
Valeurs : MQRC\_NONE:

### **ParameterCount**

Description : Nombre de structures de paramètres. Cette zone indique le nombre de structures de paramètres qui suivent la structure MQCFH. Une structure de groupe (MQCFGR) et ses structures de paramètres incluses sont comptées comme une seule structure.

Type de données : MQLONG:

Valeurs : 1 ou supérieur

### **Données de message de trace d'activité d'application**

Immédiatement après l'en-tête PCF se trouve un ensemble de paramètres décrivant l'intervalle de temps de la trace d'activité. Ces paramètres indiquent également la séquence de messages en cas d'écriture de messages. L'ordre et le nombre de zones suivant l'en-tête ne sont pas garantis, ce qui permet d'ajouter des informations supplémentaires à l'avenir.

Nom du message : Message de trace d'activité.

---

File d'attente système: SYSTEM.ADMIN.TRACE.ACTIVITY.QUEUE.

### **QueueManager**

Description : Nom du gestionnaire de files d'attente.

Identificateur : MQCA\_Q\_MGR\_NAME

Type de données : MQCFST

Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

### **QSGName**



Description : Nom du groupe de partage de files d'attente dont le gestionnaire de files d'attente est membre (z/OS uniquement).

Identificateur : NOM\_QSG\_MQCA

Type de données : MQCFST

Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

### **HostName**

Description : Nom d'hôte de la machine sur laquelle s'exécute le gestionnaire de files d'attente.

Identificateur : MQCACF\_NOM\_HÔTE

Type de données : MQCFST

### **IntervalStartDate**

Description : Date de début de la période de surveillance.

Identificateur : DATE MQCAMO\_START\_DATE

Type de données : MQCFST

Longueur maximale : LONGUEUR\_DATE\_MQ



### ***IntervalStartTime***

Description : Heure de début de la période de surveillance.  
Identificateur : HEURE DE DEMARRAGE MQCAMO\_DEMARRAGE  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE

### ***IntervalEndDate***

Description : Date de fin de la période de surveillance.  
Identificateur : DATE MQCAMO\_FIN  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_DATE\_MQ

### ***IntervalEndTime***

Description : Heure de fin de la période de surveillance.  
Identificateur : Heure\_END\_MQCAMO  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_MQ\_ATTENTE

### ***CommandLevel***

Description : Niveau de commande IBM MQ .  
Identificateur : MQIA\_NIVEAU DE COMMANDE  
Type de données : MQCFIN

### ***SeqNumber***

Description : Le numéro de séquence est normalement zéro. Cette valeur est incrémentée pour chaque enregistrement suivant pour les connexions à exécution longue.  
Identificateur : NUMÉRO\_SÉQUENCE\_MQIACF\_XX\_ENCODE\_CASE\_CAPS\_LOCK\_OFF  
Type de données : MQCFIN

### ***ApplicationName***

Description : Nom de l'application (nom du programme).  
Identificateur : NOM\_APPL\_MQCACF  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_APPL\_MQ\_

### ***ApplClass***

Description : Type d'application qui a effectué l'activité. Valeurs possibles: MQAT\_ \*  
Identificateur : TYPE\_APPL\_MQAI  
Type de données : MQCFIN

### ***ApplicationPid***

Description : ID processus du système d'exploitation de l'application.  
Identificateur : ID\_PROCESSUS\_MQIACF\_ID  
Type de données : MQCFIN

### ***UserId***

Description : Contexte d'ID utilisateur de l'application.  
Identificateur : MQCACF\_IDENTIFICATEUR\_UTILISATEUR  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_ID\_UTILISATEUR

### ***APICallerType***

Description : Type de l'application. Valeurs possibles: MQXACT\_EXTERNAL ou MQXACT\_INTERNAL.  
Identificateur : MQIACF\_API\_CALLER\_TYPE  
Type de données : MQCFIN

### ***Environment***

Description : Environnement d'exécution de l'application. Valeurs possibles: MQXE\_\*  
Identificateur : ENVIRONNEMENT MQIACF\_API  
Type de données : MQCFIN

### ***ChannelName***

Description : Nom de canal associé à la connexion. Ce paramètre est renvoyé uniquement lorsque la valeur du paramètre Environnement est MQXE\_MCA ou MQXE\_MCA\_SVRCONN.  
Identificateur : MQCACH\_CHANNEL\_NAME  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_CANAL\_MQ\_CANAL

### ***ConnectionName***

Description : Nom de connexion réseau associé à la connexion. Ce paramètre est renvoyé uniquement lorsque la valeur du paramètre Environnement est MQXE\_MCA ou MQXE\_MCA\_SVRCONN.  
Identificateur : MQCACH\_NOM\_CONNEXION  
Type de données : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_CONN\_MQ\_

### ***ChannelType***

Description : Type de canal associé à la connexion. Ce paramètre est renvoyé uniquement lorsque la valeur du paramètre Environnement est MQXE\_MCA ou MQXE\_MCA\_SVRCONN. Valeurs possibles: MQCHT\_\*

Identificateur : TYPE MQIACH\_CHANNEL\_TYPE

Type de données : MQCFIN

### **RemoteProduct**

Description : Identificateur de produit distant associé à la connexion. Ce paramètre est renvoyé uniquement lorsque la valeur du paramètre Environnement est MQXE\_MCA ou MQXE\_MCA\_SVRCONN.

Identificateur : MQCACH\_PRODUIST\_DISTANT

Type de données : MQCFST

Longueur maximale : MQ\_LONGUEUR\_PRODUIST\_DISTANT

### **RemoteVersion**

Description : Version du produit distant associée à la connexion. Ce paramètre est renvoyé uniquement lorsque la valeur du paramètre Environnement est MQXE\_MCA ou MQXE\_MCA\_SVRCONN.

Identificateur : VERSION\_DISTANT\_CACHE

Type de données : MQCFST

Longueur maximale : LONGUEUR\_VERSION\_DISTANT\_MQ\_DISTANTE

### **FunctionName**

Description : Nom de la dernière fonction de haut niveau lancée par l'unité d'exécution initiale.

Identificateur : MQCACF\_FONCTION\_APPL\_MQCACF

Type de données : MQCFST

### **FunctionType**

Description : Type de la dernière fonction de haut niveau lancée par l'unité d'exécution initiale. Valeurs possibles: MQFUN\_\*

Identificateur : TYPE\_FONCTION\_APPL\_MQIACF\_XX\_ENCODE\_CASE\_CAPS\_LOCK\_OFF

Type de données : MQCFIN

### **Detail**

Description : Niveau de détail enregistré pour la connexion. Valeurs possibles: 1=LOW 2=MEDIUM 3=HIGH

Identificateur : MQIACF\_TRACE\_DETAIL

Type de données : MQCFIN

### **TraceDataLength**

Description : Longueur des données de message (en octets) tracées pour cette connexion.

Identificateur : LONGUEUR\_DONNÉES\_TRACE\_MQIACF\_

Type de données : MQCFIN

### **PointerSize**

Description : Longueur (en octets) des pointeurs sur la plateforme que l'application exécute (pour faciliter l'interprétation des structures binaires).

Identificateur : TAILLE MQIACF\_POINTER\_taille

Type de données : MQCFIN

### **Platform**

Description : Plateforme sur laquelle le gestionnaire de files d'attente s'exécute. Valeurs possibles: MQPL\_\*

Identificateur : MQIA\_PLATEFORME

Type de données : MQCFIN

### **Paramètres de variable pour les opérations MQI d'activité d'application**

La structure MQCFGR des données d'activité de l'application est suivie de l'ensemble de paramètres PCF qui correspond à l'opération en cours. Les paramètres de chaque opération sont définis dans la section suivante.

Le niveau de trace indique le niveau de granularité de trace requis pour que les paramètres soient inclus dans la trace. Les valeurs de niveau de trace possibles sont les suivantes:

#### 1. Bas

Le paramètre est inclus lorsque le traçage d'activité "faible", "moyenne" ou "élevée" est configuré pour une application. Ce paramètre signifie qu'un paramètre est toujours inclus dans le groupe AppActivityData pour l'opération. Cet ensemble de paramètres est suffisant pour tracer les appels MQI émis par une application et pour voir s'ils ont abouti.

#### 2. Moyen

Le paramètre est uniquement inclus dans le groupe AppActivityData pour l'opération lorsque la fonction de trace de l'activité "moyenne" ou "élevée" est configurée pour une application. Cet ensemble de paramètres ajoute des informations sur les ressources, par exemple, les noms de file d'attente et de rubrique utilisés par l'application.

#### 3. Elevé

Le paramètre est uniquement inclus dans le groupe AppActivityData pour l'opération lorsque le traçage d'activité "élevé" est configuré pour une application. Cet ensemble de paramètres inclut des vidages mémoire des structures transmises aux fonctions MQI et XA. Pour cette raison, il contient des informations supplémentaires sur les paramètres utilisés dans les appels MQI et XA. Les vidages mémoire de la structure sont des copies peu profondes des structures. Pour éviter les tentatives erronées de déréréférencement des pointeurs, les valeurs de pointeur dans les structures sont définies sur NULL.

**Remarque :** La version de la structure faisant l'objet d'un vidage n'est pas nécessairement identique à la version utilisée par une application. La structure peut être modifiée par un exit de croisement d'API, par le code de trace d'activité ou par le gestionnaire de files d'attente. Un gestionnaire de files d'attente peut modifier une structure vers une version ultérieure, mais le gestionnaire de files d'attente ne la remplace jamais par une version antérieure de la structure. Pour ce faire, vous risquez de perdre des données.

### **MQBACK**

L'application a démarré la fonction MQBACK MQI

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération

Paramètre PCF: CODE COMP\_MQIACF

Niveau de trace : 1  
Tapez MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Tapez MQCFIN

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  
  
La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  
  
**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.  
  
Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez MQCFIN64

### **MQBEGIN**

L'application a démarré la fonction MQBEGIN MQI

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Tapez MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Tapez MQCFIN

### **MQBO**

Description : Structure des options MQBEGIN. Ce paramètre n'est pas inclus si un pointeur NULL est utilisé dans l'appel MQBEGIN.  
  
Paramètre PCF: MQBACF\_MQBO\_STRUCT  
Niveau de trace : 3  
Tapez MQCFBS  
Longueur : Longueur en octets de la structure MQBO.

### ***QMGrOpDuration***

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION

Niveau de trace : 2

Tapez MQCFIN64

### ***MQCALLBACK***

L'application a démarré la fonction MQCALLBACK

### ***ObjectHandle***

Description : Descripteur d'objet

Paramètre PCF: MQIACF\_HOBJ

Niveau de trace : 1

Tapez MQCFIN

### ***CallType***

Description : Pourquoi la fonction a été appelée. Une des valeurs MQCBCT\_\*

Paramètre PCF: MQIACF\_TYPE\_APPEL

Niveau de trace : 1

Tapez MQCFIN

### ***MsgBuffer***

Description : Données de message.

Paramètre PCF: DONNÉES\_MESSAGE\_MQBACF\_MESSAGE

Niveau de trace : 1

Tapez MQCFBS

Longueur : La longueur est régie par le paramètre TRACEDATA () défini dans la configuration APPTTRACE. Si TRACEDATA=NONE, ce paramètre est omis.

### ***MsgLength***

Description : Longueur du message. (extrait de la zone DataLength dans la structure MQCBC).

Paramètre PCF: LONGUE\_MSG\_MQIACF\_MESSAGE

Niveau de trace : 1

Tapez MQCFIN

### **HighResTime**

Description : Heure de fonctionnement en microsecondes depuis minuit, le 1st 1970 (UTC)  
**Remarque :** La précision de cette minuterie varie en fonction de la prise en charge de la plateforme pour une minuterie à haute résolution

Paramètre PCF: MQIAMO64\_HIGHRES\_TIME  
Niveau de trace : 2  
Tapez MQCFIN64

### **ReportOptions**

Description : Options des messages de rapport  
Paramètre PCF: RAPPORT MQIACF\_RAPPORT  
Niveau de trace : 2  
Tapez MQCFIN

### **MsgType**

Description : Type de message  
Paramètre PCF: TYPE MQIACF\_MSG\_  
Niveau de trace : 2  
Tapez MQCFIN

### **Expiry**

Description : Durée de vie des messages  
Paramètre PCF: EXPIRATION\_MQIACF\_EXPIRATION  
Niveau de trace : 2  
Tapez MQCFIN

### **Format**

Description : Nom de format des données de message  
Paramètre PCF: MQCACH\_FORMAT\_NAME  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : LONGUEUR\_FORMAT\_MQ

### **Priority**

Description : Priorité de message  
Paramètre PCF: PRIO\_MQIACF\_ACF  
Niveau de trace : 2  
Tapez MQCFIN

### **Persistence**

Description : Persistance des messages

Paramètre PCF: PERSISTENCE MQIACF\_PERSISTANCE  
Niveau de trace : 2  
Tapez MQCFIN

### ***MsgId***

Description : Identificateur de message  
Paramètre PCF: ID\_MSG\_MQBACF  
Niveau de trace : 2  
Tapez MQCFBS  
Longueur : LONGUEUR\_MSG\_MQ\_ID

### ***CorrelId***

Description : Identificateur de corrélation  
Paramètre PCF: ID\_CORREL\_MQBACF  
Niveau de trace : 2  
Tapez MQCFBS  
Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### ***ObjectName***

Description : Nom de l'objet ouvert.  
Paramètre PCF: MQCACF\_NOM\_OBJET  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***ResolvedQName***

Description : Nom local de la file d'attente à partir de laquelle le message a été extrait.  
Paramètre PCF: MQCACF\_RESOLVED\_Q\_NAME  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***ReplyToQueue***

Description : LONGUEUR\_NOM\_Q\_MQ\_  
Paramètre PCF: MQCACF\_RÉPONSE\_À\_Q  
Niveau de trace : 2  
Tapez MQCFST

### ***ReplyToQMgr***

Description : MQ\_Q\_MGR\_NAME\_LENGTH  
Paramètre PCF: MQCACF\_REPLY\_TO\_Q\_MGR



Niveau de trace : 2  
Tapez MQCFST

### ***CodedCharSetId***

Description : Identificateur de jeu de caractères des données de message  
Paramètre PCF: MQIA\_CODED\_CHAR\_SET\_ID  
Niveau de trace : 2  
Tapez MQCFIN

### ***Encoding***

Description : Codage numérique des données de message.  
Paramètre PCF: CODAGE\_MQIACF\_CODAGE  
Niveau de trace : 2  
Tapez MQCFIN

### ***PutDate***

Description : LONGUEUR\_DATE\_PUT\_MQ\_  
Paramètre PCF: DATE MQCACF\_PUT\_DATE  
Niveau de trace : 2  
Tapez MQCFST

### ***PutTime***

Description : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG  
Paramètre PCF: MQCACF\_HEURE  
Niveau de trace : 2  
Tapez MQCFST

### ***ResolvedQName***

Description : Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT\_Q.  
Paramètre PCF: MQCACF\_RESOLVED\_LOCAL \_NOM\_QD  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Description : Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT\_TOPIC.  
Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : La longueur varie.

### ***ResolvedType***

Description :	Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT_Q, MQOT_TOPIC ou MQOT_NONE.
Paramètre PCF:	MQIACF_RESOLVED_TYPE
Niveau de trace :	2
Tapez	MQCFIN

### ***PolicyName***

Description :	Nom de la règle qui a été appliquée à ce message. <b>Remarque :</b> Messages protégés AMS uniquement
Paramètre PCF:	MQCA_NOM_REGLE
Niveau de trace :	2
Tapez	MQCFST
Longueur :	LONGUEUR_NOM_OBJET_MQ_

### ***XmitqMsgId***

Description :	ID du message dans l'en-tête de la file d'attente de transmission. <b>Remarque :</b> Uniquement lorsque le format est MQFMT_XMIT_Q_HEADER
Paramètre PCF:	MQBACF_XQH_ID_MSG
Niveau de trace :	2
Tapez	MQCFBS
Longueur :	LONGUEUR_MSG_MQ_ID

### ***XmitqCorrelId***

Description :	ID de corrélation du message dans l'en-tête de la file d'attente de transmission. <b>Remarque :</b> Uniquement lorsque le format est MQFMT_XMIT_Q_HEADER
Paramètre PCF:	MQBACF_XQH_ID_CORRÉLATION
Niveau de trace :	2
Tapez	MQCFBS
Longueur :	LONGUEUR_ID_CORREL_MQ_

### ***XmitqPutTime***

Description :	Heure d'insertion du message dans l'en-tête de la file d'attente de transmission. <b>Remarque :</b> Uniquement lorsque le format est MQFMT_XMIT_Q_HEADER
Paramètre PCF:	MQCACF_XQH_HEURE
Niveau de trace :	2
Tapez	MQCFST
Longueur :	LONGUEUR_TEMPS_PUT_MQ_LONG

### ***XmitqPutDate***

Description :	Date d'insertion du message dans l'en-tête de la file de transmission. <b>Remarque :</b> Uniquement lorsque le format est MQFMT_XMIT_Q_HEADER
Paramètre PCF:	DATE MQCACF_XQH_PUT_DATE
Niveau de trace :	2
Tapez	MQCFST
Longueur :	LONGUEUR_DATE_PUT_MQ_

### ***XmitqRemoteQName***

Description :	Destination de file d'attente éloignée du message dans l'en-tête de la file d'attente de transmission. <b>Remarque :</b> Uniquement lorsque le format est MQFMT_XMIT_Q_HEADER
Paramètre PCF:	MQCACF_XQH_REMOTE_Q_Name
Niveau de trace :	2
Tapez	MQCFST
Longueur :	LONGUEUR_NOM_Q_MQ_

### ***XmitqRemoteQMGr***

Description :	ID du message dans l'en-tête de la file d'attente de transmission. <b>Remarque :</b> Uniquement lorsque le format est MQFMT_XMIT_Q_HEADER
Paramètre PCF:	MQCACF_XQH_REMOTE_Q_MGR
Niveau de trace :	2
Tapez	MQCFST
Longueur :	LONGUEUR_MSG_MQ_ID

### ***MsgDescStructure***

Description :	Structure MQMD. Ce paramètre est omis si un MQGMO version 4 a été utilisé pour demander le renvoi d'un descripteur de message à la place d'un MQMD
Paramètre PCF:	MQBACF_MQMD_STRUCT
Niveau de trace :	3
Tapez	MQCFBS
Longueur :	Longueur en octets de la structure MQMD (la taille réelle dépend de la version de la structure)

### ***GetMsgOptsStructure***

Description :	Structure MQGMO.
Paramètre PCF:	MQBACF_MQGMO_STRUCT
Niveau de trace :	3
Tapez	MQCFBS
Longueur :	Longueur en octets de la structure MQGMO (la taille réelle dépend de la version de la structure)

### ***MQCBCContextStructure***

Description :	Structure MQCBC.
Paramètre PCF:	MQBACF_MQCBC_STRUCT
Niveau de trace :	3
Tapez	MQCFBS
Longueur :	Longueur en octets de la structure MQCBC (la taille réelle dépend de la version de la structure)

### ***QMGrOpDuration***

Description :	Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  <b>Remarque :</b> La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.
Paramètre PCF:	MQIAMO64_QMGR_OP_DURATION
Niveau de trace :	2
Tapez	MQCFIN64

### ***MQCB***

L'application a démarré la fonction MQI de rappel de gestion

### ***CallbackOperation***

Description :	Opération de la fonction de rappel de gestion. Défini sur l'une des valeurs MQOP_*
Paramètre PCF:	Opération MQIACF_MQCB_OPERATION
Niveau de trace :	1
Tapez	MQCFIN

### ***CallbackType***

Description :	Type de la fonction de rappel (zoneCallbackType de la structure MQCBD). Défini sur l'une des valeurs MQCBT_*
Paramètre PCF:	TYPE MQIACF_MQCB_
Niveau de trace :	1
Tapez	MQCFIN

### ***CallbackOptions***

Description :	Options de rappel. Défini sur l'une des valeurs MQCBDO_*
Paramètre PCF:	OPTIONS MQIACF_MQCB_
Niveau de trace :	1
Tapez	MQCFIN

### ***CallbackFunction***

Description :	Pointeur vers la fonction de rappel si elle est démarrée en tant qu'appel de fonction.
Paramètre PCF:	MQBACF_MQCB_FONCTION
Niveau de trace :	1
Tapez	MQCFBS
Longueur :	Taille de MQPTR

### ***CallbackName***

Description :	Nom de la fonction de rappel si elle est démarrée en tant que programme lié dynamiquement.
Paramètre PCF:	NOM_MQCACF_MQCB_
Niveau de trace :	1
Tapez	MQCFST
Longueur :	Taille de MQCHAR128

### ***ObjectHandle***

Description :	Descripteur d'objet
Paramètre PCF:	MQIACF_HOBJ
Niveau de trace :	1
Tapez	MQCFIN

### ***MaxMsgLength***

Description :	Longueur maximale des messages. Défini sur un entier ou sur la valeur spéciale MQCBD_FULL_MSG_LENGTH
Paramètre PCF:	LONGUEUR_MSG_MAX_MQIACH_MQI
Niveau de trace :	2
Tapez	MQCFIN

### ***CompCode***

Description :	Code achèvement indiquant le résultat de l'opération
Paramètre PCF:	CODE_COMP_MQIACF
Niveau de trace :	1
Tapez	MQCFIN

### ***Reason***

Description :	Résultat du code anomalie de l'opération
Paramètre PCF:	MQIACF_CODE_RAISON
Niveau de trace :	1
Tapez	MQCFIN

### ***ResolvedQName***

Description :	Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT_Q.
Paramètre PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Niveau de trace :	2
Tapez	MQCFST
Longueur :	MQ_Q_NAME_LENGTH.

### ***ResObjectString***

Description :	Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT_TOPIC.
Paramètre PCF:	MQCACF_RESOLVED_OBJECT_STRING
Niveau de trace :	2
Tapez	MQCFST
Longueur :	La longueur varie.

### ***ResolvedType***

Description :	Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT_Q, MQOT_TOPIC ou MQOT_NONE.
Paramètre PCF:	MQIACF_RESOLVED_TYPE
Niveau de trace :	2
Tapez	MQCFIN

### ***Callback DescriptorStructure***

Description :	Structure MQCBD. Ce paramètre est omis si une valeur MQCBC NULL est transmise à l'appel MQCB.
Paramètre PCF:	MQBACF_MQCBD_STRUCT
Niveau de trace :	3
Tapez	MQCFBS
Longueur :	Longueur en octets de la structure MQCBC

### ***MsgDescStructure***

Description :	Structure MQMD. Le paramètre de structure MsgDesc est omis si une valeur MQMD NULL est transmise à l'appel MQCB.
Paramètre PCF:	MQBACF_MQMD_STRUCT
Niveau de trace :	3
Tapez	MQCFBS
Longueur :	Longueur en octets de la structure MQMD (la taille réelle dépend de la version de la structure)

### ***GetMsgOptsStructure***

Description :	Structure MQGMO. Ce paramètre est omis si une valeur MQGMO NULL est transmise à l'appel MQCB.
---------------	---

Paramètre PCF: MQBACF\_MQGMO\_STRUCT  
Niveau de trace : 3  
Tapez MQCFBS  
Longueur : Longueur en octets de la structure MQGMO (la taille réelle dépend de la version de la structure)

### ***QMGrOpDuration***

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez MQCFIN64

### ***MQCLOSE***

L'application a démarré la fonction MQCLOSE MQI

### ***ObjectHandle***

Description : Descripteur d'objet  
Paramètre PCF: MQIACF\_HOBJ  
Niveau de trace : 1  
Tapez MQCFIN

### ***CloseOptions***

Description : Options de fermeture  
Paramètre PCF: OPTIONS MQIACF\_CLOSE\_OPTIONS  
Niveau de trace : 1  
Tapez MQCFIN

### ***CompCode***

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Tapez MQCFIN

### ***Reason***

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Tapez MQCFIN

### **ResolvedQName**

Description :	Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT_Q.
Paramètre PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Niveau de trace :	2
Tapez	MQCFST
Longueur :	MQ_Q_NAME_LENGTH.

### **ResObjectString**

Description :	Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT_TOPIC.
Paramètre PCF:	MQCACF_RESOLVED_OBJECT_STRING
Niveau de trace :	2
Tapez	MQCFST
Longueur :	La longueur varie.

### **ResolvedType**

Description :	Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT_Q, MQOT_TOPIC ou MQOT_NONE.
Paramètre PCF:	MQIACF_RESOLVED_TYPE
Niveau de trace :	2
Tapez	MQCFIN

### **QMgrOpDuration**

Description :	Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  <b>Remarque :</b> La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.
Paramètre PCF:	MQIAMO64_QMGR_OP_DURATION
Niveau de trace :	2
Tapez	MQCFIN64

### **MQCMIT**

L'application a démarré la fonction MQI MQCMIT

### **CompCode**

Description :	Code achèvement indiquant le résultat de l'opération
Paramètre PCF:	CODE COMP_MQIACF
Niveau de trace :	1
Tapez	MQCFIN



### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Tapez MQCFIN

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez MQCFIN64

### **MQCONN et MQCONNX**

L'application a démarré la fonction MQCONN ou MQCONNX MQI

### **ConnectionId**

Description : L'ID connexion s'il est disponible ou MQCONNID\_NONE s'il ne l'est pas  
Paramètre PCF: ID\_CONNEXION\_MQBACF  
Niveau de trace : 1  
Type : MQCFBS  
Longueur maximale : LONGUEUR\_ID\_CONNEXION\_MQM

### **QueueManagerName**

Description : Nom (non résolu) du gestionnaire de files d'attente utilisé dans l'appel MQCONN (X)  
Paramètre PCF: MQCA\_Q\_MGR\_NAME  
Niveau de trace : 1  
Type : MQCFST  
Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE\_COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

### **ConnectOptions**

Description : Options de connexion dérivées des valeurs MQCNO\_\*  
**Remarque :** MQCONNX uniquement  
Paramètre PCF: MQIACF\_CONNECT\_OPTIONS  
Niveau de trace : 2  
Type : MQCFIN

### **ConnectionOptionsStructure**

Description : Structure MQCNO.  
**Remarque :** MQCONNX uniquement)  
Paramètre PCF: MQBACF\_MQCNO\_STRUCT  
Niveau de trace : 3  
Type : MQCFBS  
Longueur maximale : Longueur en octets de la structure MQCNO (la taille réelle dépend de la version de la structure)

### **ChannelDefinitionStructure**

Description : Structure MQCD.  
**Remarque :** Connexions client uniquement  
Paramètre PCF: MQBACF\_MQCD\_STRUCT  
Niveau de trace : 3  
Type : MQCFBS  
Longueur maximale : Longueur en octets de la structure MQCD (la taille réelle dépend de la version de la structure)

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  
La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  
**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.  
Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez : MQCFIN64

## *MQCTL*

L'application a démarré la fonction MQI MQCTL

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

### **CtlOperation**

Description : L'une des valeurs MQOP\_\*  
Paramètre PCF: OPERATION MQIACF\_CTL\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez MQCFIN64

## *MQDISC*

L'application a démarré la fonction MQDISC MQI

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON

Niveau de trace : 1  
Type : MQCFIN

### *MQGET*

L'application a démarré la fonction MQGET MQI

#### **ObjectHandle**

Description : Descripteur d'objet  
Paramètre PCF: MQIACF\_HOBJ  
Niveau de trace : 1  
Type : MQCFIN

#### **GetOptions**

Description : Options d'obtention de MQGMO.Options  
Paramètre PCF: OPTIONS MQIACF\_GET\_OPTIONS  
Niveau de trace : 1  
Type : MQCFIN

#### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

#### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

#### **MsgBuffer**

Description : Données de message. Si TRACEDATA=NONE, ce paramètre est omis  
Paramètre PCF: DONNÉES\_MESSAGE\_MQBACF\_MESSAGE  
Niveau de trace : 1  
Type : MQCFBS  
Longueur maximale : La longueur est régie par le paramètre TRACEDATA () défini dans la configuration APPTRACE. (Inclus dans le message de trace en tant que MQIACF\_TRACE\_DATA\_LENGTH).

#### **MsgLength**

Description : Longueur du message.  
Paramètre PCF: LONGUE\_MSG\_MQIACF\_MESSAGE  
Niveau de trace : 1

Type : MQCFIN

### **HighResTime**

Description : Heure de fonctionnement en microsecondes depuis minuit, le 1er janvier 1970 (UTC)

**Remarque :** La précision de cette minuterie varie en fonction de la prise en charge de la plateforme pour une minuterie à haute résolution

Paramètre PCF: MQIAMO64\_HIGHRES\_TIME

Niveau de trace : 2

Type : MQCFIN64

### **BufferLength**

Description : Longueur de la mémoire tampon fournie par l'application

Paramètre PCF: LONGUEUR\_BUFFER\_MQIACF

Niveau de trace : 2

Type : MQCFIN

### **ObjectName**

Description : Nom de l'objet ouvert

Paramètre PCF: MQCACF\_NOM\_OBJET

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **ResolvedQName**

Description : Nom local de la file d'attente à partir de laquelle le message a été extrait.

Paramètre PCF: MQCACF\_RESOLVED\_Q\_NAME

Niveau de trace : 2

Type : MQCFST

Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_

### **ReportOptions**

Description : Options de rapport de message

Paramètre PCF: RAPPORT MQIACF\_RAPPORT

Niveau de trace : 2

Type : MQCFIN

### **MsgType**

Description : Type de message

Paramètre PCF: TYPE MQIACF\_MSG\_

Niveau de trace : 2

Type : MQCFIN

### ***Expiry***

Description : Durée de vie des messages  
Paramètre PCF: EXPIRATION\_MQIACF\_EXPIRATION  
Niveau de trace : 2  
Type : MQCFIN

### ***Format***

Description : Nom de format des données de message  
Paramètre PCF: MQCACH\_FORMAT\_NAME  
Niveau de trace : 2  
Type : MQCFST  
Longueur maximale : LONGUEUR\_FORMAT\_MQ

### ***Priority***

Description : Priorité de message  
Paramètre PCF: PRIO\_MQIACF\_ACF  
Niveau de trace : 2  
Type : MQCFIN

### ***Persistence***

Description : Persistence des messages  
Paramètre PCF: PERSISTENCE\_MQIACF\_PERSISTANCE  
Niveau de trace : 2  
Type : MQCFIN

### ***MsgId***

Description : Identificateur de message  
Paramètre PCF: ID\_MSG\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS  
Longueur maximale : LONGUEUR\_MSG\_MQ\_ID

### ***CorrelId***

Description : Identificateur de corrélation  
Paramètre PCF: ID\_CORREL\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS

Longueur maximale : LONGUEUR\_ID\_CORREL\_MQ\_

### ***ReplyToQueue***

Description :

Paramètre PCF: MQCACF\_RÉPONSE\_À\_Q

Niveau de trace : 2

Type : MQCFST

Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_

### ***ReplyToQMGr***

Description :

Paramètre PCF: MQCACF\_REPLY\_TO\_Q\_MGR

Niveau de trace : 2

Type : MQCFST

Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

### ***CodedCharSetId***

Description : Identificateur de jeu de caractères des données de message

Paramètre PCF: MQIA\_CODED\_CHAR\_SET\_ID

Niveau de trace : 2

Type : MQCFIN

### ***Encoding***

Description : Codage numérique des données de message.

Paramètre PCF: CODAGE\_MQIACF\_CODAGE

Niveau de trace : 2

Type : MQCFIN

### ***PutDate***

Description :

Paramètre PCF: DATE MQCACF\_PUT\_DATE

Niveau de trace : 2

Type : MQCFST

Longueur maximale : LONGUEUR\_DATE\_PUT\_MQ\_

### ***PutTime***

Description :

Paramètre PCF: MQCACF\_HEURE

Niveau de trace : 2

Type : MQCFST  
Longueur maximale : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG

### ***ResolvedQName***

Description : Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT\_Q.  
Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Description : Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT\_TOPIC.  
Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : La longueur varie.

### ***ResolvedType***

Description : Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT\_Q, MQOT\_TOPIC ou MQOT\_NONE.  
Paramètre PCF: MQIACF\_RESOLVED\_TYPE  
Niveau de trace : 2  
Tapez MQCFIN

### ***PolicyName***

Description : Nom de la règle qui a été appliquée à ce message.  
**Remarque :** Messages protégés AMS uniquement  
Paramètre PCF: MQCA\_NOM\_REGLE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_OBJET\_MQ\_

### ***XmitqMsgId***

Description : ID du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER  
Paramètre PCF: MQBACF\_XQH\_ID\_MSG  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_MSG\_MQ\_ID



### ***XmitqCorrelId***

Description : ID de corrélation du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQBACF\_XQH\_ID\_CORRÉLATION

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### ***XmitqPutTime***

Description : Heure d'insertion du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_HEURE

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG

### ***XmitqPutDate***

Description : Date d'insertion du message dans l'en-tête de la file de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: DATE MQCACF\_XQH\_PUT\_DATE

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_DATE\_PUT\_MQ\_

### ***XmitqRemoteQName***

Description : Destination de file d'attente éloignée du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_REMOTE\_Q\_NAME

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***XmitqRemoteQMgr***

Description : Destination du gestionnaire de files d'attente éloignées du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_REMOTE\_Q\_MGR

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***MsgDescStructure***

Description : Structure MQMD.  
Paramètre PCF: MQBACF\_MQMD\_STRUCT  
Niveau de trace : 3  
Type : MQCFBS  
Longueur maximale : Longueur en octets de la structure MQMD (la taille réelle dépend de la version de la structure)

### ***GetMsgOptsStructure***

Description : Structure MQGMO.  
Paramètre PCF: MQBACF\_MQGMO\_STRUCT  
Niveau de trace : 3  
Type : MQCFBS  
Longueur maximale : Longueur en octets de la structure MQGMO (la taille réelle dépend de la version de la structure)

### ***QMGrOpDuration***

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez MQCFIN64

### ***MQINQ***

L'application a démarré la fonction MQINQ MQI

### ***ObjectHandle***

Description : Descripteur d'objet  
Paramètre PCF: MQIACF\_HOBJ  
Niveau de trace : 1  
Type : MQCFIN

### ***CompCode***

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE\_COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

### **SelectorCount**

Description : Nombre de sélecteurs fournis dans le tableau Sélecteurs.  
Paramètre PCF: MQIACF\_SELECTOR\_COUNT  
Niveau de trace : 2  
Type : MQCFIN

### **Selectors**

Description : Liste des attributs (entiers ou caractères) dont les valeurs doivent être renvoyées par MQINQ.  
Paramètre PCF: Sélecteurs MQIACF\_SELECTORS  
Niveau de trace : 2  
Type : MQCFIL

### **ResolvedQName**

Description : Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT\_Q.  
Paramètre PCF: MQCACF\_RESOLVED\_Q\_NAME  
Niveau de trace : 2  
Type : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_

### **ResObjectString**

Description : Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT\_TOPIC.  
Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING  
Niveau de trace : 2  
Type : MQCFST  
Longueur maximale : La longueur varie

### **ResolvedType**

Description : Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT\_Q, MQOT\_TOPIC ou MQOT\_NONE.  
Paramètre PCF: MQIACF\_RESOLVED\_TYPE  
Niveau de trace : 2  
Type : MQCFIN

### **IntAttrCount**

Description :	Nombre d'attributs de type entier renvoyés par l'opération d'interrogation
Paramètre PCF:	MQIACF_INTATTR_COUNT
Niveau de trace :	3
Type :	MQCFIN

### **IntAttr**

Description :	Valeurs d'attribut entières renvoyées par l'opération d'interrogation. Ce paramètre n'est présent que si IntAttrCount est > 0 lorsque MQINQ est renvoyé.
Paramètre PCF:	MQIACF_INT_ATTRS
Niveau de trace :	3
Type :	MQCFIL

### **CharAttr**

Description :	Attributs de caractère renvoyés par l'opération d'interrogation. Les valeurs sont concaténées. Ce paramètre est inclus uniquement si la longueur CharAttr est > 0 lorsque MQINQ est renvoyé.
Paramètre PCF:	MQCACF_CHAR_ATTRS
Niveau de trace :	3
Type :	MQCFST

### **QMGrOpDuration**

Description :	Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  <b>Remarque :</b> La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.
Paramètre PCF:	MQIAMO64_QMGR_OP_DURATION
Niveau de trace :	2
Tapez	MQCFIN64

### **MQOPEN**

L'application a démarré la fonction MQOPEN MQI

### **ObjectType**

Description :	Type d'objet transmis dans MQOT.ObjectType
Paramètre PCF:	TYPE_OBS_MQIACF
Niveau de trace :	1
Type :	MQCFIN

### **ObjectName**

Description :	Nom de l'objet transmis à l'appel MQI avant toute tentative de résolution de nom de file d'attente.
---------------	---

Paramètre PCF: MQCACF\_NOM\_OBJET  
Niveau de trace : 1  
Type : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_

### ***ObjectQMgrName***

Description : Nom du gestionnaire de files d'attente d'objet transmis à l'appel MQI avant toute tentative de résolution de nom de file d'attente.  
Paramètre PCF: MQCACF\_OBJECT\_Q\_MGR\_NAME  
Niveau de trace : 1  
Type : MQCFST  
Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

### ***ObjectHandle***

Description : Descripteur d'objet  
Paramètre PCF: MQIACF\_HOBJ  
Niveau de trace : 1  
Type : MQCFIN

### ***CompCode***

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE\_COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### ***Reason***

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

### ***OpenOptions***

Description : Options utilisées pour ouvrir l'objet  
Paramètre PCF: OPTIONS\_OPEN\_MQIACF\_ACF  
Niveau de trace : 1  
Type : MQCFIN

### ***AlternateUserId***

Description : Inclus uniquement si MQOO\_ALTERNATE\_USER\_AUTHORITY est spécifié  
Paramètre PCF: MQCACF\_ALTERNATE\_USERID

Niveau de trace : 2  
Type : MQCFST  
Longueur maximale : LONGUEUR\_ID\_UTILISATEUR

### ***RecsPresent***

Description : Nombre d'enregistrements de nom d'objet présents. Inclus uniquement si MQOD Version > = MQOD\_VERSION\_2  
Paramètre PCF: MQIACF\_RECS\_PRESENT  
Niveau de trace : 1  
Type : MQCFIN

### ***KnownDestCount***

Description : Nombre de files d'attente locales ouvertes avec succès uniquement si MQOD Version > = MQOD\_VERSION\_2  
Paramètre PCF: MQIACF\_KNOWN\_DEST\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

### ***UnknownDestCount***

Description : Nombre de files d'attente distantes ouvertes avec succès uniquement si MQOD Version > = MQOD\_VERSION\_2  
Paramètre PCF: MQIACF\_UNKNOWN\_DEST\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

### ***InvalidDestCount***

Description : Nombre de files d'attente dont l'ouverture a échoué uniquement si MQOD Version > = MQOD\_VERSION\_2  
Paramètre PCF: MQIACF\_INVALID\_DEST\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

### ***DynamicQName***

Description : Nom de la file d'attente dynamique transmis en entrée à l'appel MQOPEN.  
Paramètre PCF: NOM\_Q\_DYNAMIC\_MQCACF  
Niveau de trace : 2  
Type : MQCFST  
Longueur maximale : LONGUEUR\_NOM\_Q\_MQ\_

### **ResolvedLocalQName** <sup>1 2</sup>

Description :	Contient le nom de la file d'attente locale une fois la résolution de nom effectuée. (par exemple, pour les files d'attente éloignées, il s'agit du nom de la file d'attente de transmission)
Paramètre PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Niveau de trace :	2
Type :	MQCFST
Plage :	Si MQOD.Version est inférieure à MQOD_VERSION_3 et contient la valeur de MQOD.ObjectName une fois l'appel MQOPEN terminé. Si MQOD.Version est supérieure ou égale à MQOD_VERSION_3 , elle contient la valeur dans le MQOD. Zone ResolvedQName .
Longueur maximale :	LONGUEUR_NOM_Q_MQ_

### **ResolvedLocalQMgrName** <sup>1 2</sup>

Description :	Nom du gestionnaire de files d'attente local après la résolution de nom.
Paramètre PCF:	MQCACF_RESOLVED_LOCAL_Q_MGR
Niveau de trace :	2
Type :	MQCFST
Plage :	Uniquement si MQOD.Version > = MQOD_VERSION_3
Longueur maximale :	MQ_Q_MGR_NAME_LENGTH

### **ResolvedQName** <sup>1 2</sup>

Description :	Le nom de la file d'attente après la résolution de nom a été effectué.
Paramètre PCF:	MQCACF_RESOLVED_Q_NAME
Niveau de trace :	2
Type :	MQCFST
Plage :	Si MQOD.Version est inférieure à MQOD_VERSION_3 et contient la valeur de MQOD.ObjectName une fois l'appel MQOPEN terminé. Si MQOD.Version est supérieure ou égale à MQOD_VERSION_3 , elle contient la valeur dans le MQOD. Zone ResolvedQName .
Longueur maximale :	LONGUEUR_NOM_Q_MQ_

### **ResolvedQMgrName** <sup>1 2</sup>

Description :	Contient le nom du gestionnaire de files d'attente une fois la résolution de nom effectuée. Si MQOD.Version est inférieure à MQOD_VERSION_3 , elle contient la valeur du MQOD. Zone de nom ObjectQMgrune fois l'appel MQOPEN terminé. Si MQOD.Version est supérieure ou égale à MQOD_VERSION_3 , elle contient la valeur dans le MQOD. Zone de nom ResolvedQMgr.
Paramètre PCF:	MQCACF_RESOLVED_Q_MGR
Niveau de trace :	2
Type :	MQCFST

Longueur maximale : MQ\_Q\_MGR\_NAME\_LENGTH

### ***AlternateSecurityId***

Description : Autre identificateur de sécurité. Présent uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_3, MQOO\_ALTERNATE\_USER\_AUTHORITY est spécifié et MQOD.AlternateSecurityId n'est pas égal à MQSID\_NONE.

Paramètre PCF: MQBACF\_ALTERNATE\_SECURITYID

Niveau de trace : 2

Type : MQCFBS

Longueur maximale : LONGUEUR\_ID\_SÉCURITÉ\_MQ

### ***ObjectString***

Description : Nom d'objet long. Inclus uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4 et à la zone VSLength de MQOD. ObjectString est MQVS\_NULL\_TERMINATED ou supérieure à zéro.

Paramètre PCF: MQCACF\_OBJECT\_CHAINE

Niveau de trace : 2

Type : MQCFST

Longueur maximale : La longueur varie.

### ***SelectionString***

Description : Chaîne de sélection. Inclus uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4 et à la zone VSLength de MQOD. SelectionString est MQVS\_NULL\_TERMINATED ou supérieur à zéro.

Paramètre PCF: CHAÎNE\_SÉLECTION\_MQCACF

Niveau de trace : 2

Type : MQCFST

Longueur maximale : La longueur varie.

### ***ResObjectString***

Description : Le nom d'objet long après le gestionnaire de files d'attente résout le nom fourni dans la zone ObjectName . Inclus uniquement pour les rubriques et les alias de file d'attente qui font référence à un objet de rubrique si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4 et VSLength est MQVS\_NULL\_TERMINATED ou supérieure à zéro.

Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING

Niveau de trace : 2

Type : MQCFST

Longueur maximale : La longueur varie.



### **ResolvedType**

Description : Type de l'objet résolu (de base) en cours d'ouverture. Inclus uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4. Les valeurs possibles sont MQOT\_Q, MQOT\_TOPIC ou MQOT\_NONE.

Paramètre PCF: MQIACF\_RESOLVED\_TYPE

Niveau de trace : 2

Type : MQCFIN

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION

Niveau de trace : 2

Tapez MQCFIN64

### *Structure d'en-tête de groupe PCF de la liste de distribution d'activité d'application*

Si la fonction MQOPEN ouvre une liste de distribution, les paramètres MQOPEN incluent un groupe PCF AppActivityDistList pour chacune des files d'attente de la liste de distribution jusqu'au nombre de structures numérotées dans RecsPresent. Le groupe PCF ap-pActivityDistList combine les informations des structures MQOR et MQRR pour identifier le nom de la file d'attente et indiquer le résultat de l'opération d'ouverture sur la file d'attente. Un groupe AppActivityDistList commence toujours par la structure MQCFGR suivante:

<b>MQCFGR, zone</b>	<b>Valeur</b>	<b>Description</b>
Tapez	GROUPE MQCF	
StrucLength	Longueur en octets de la structure MQCFGR	
Paramètre	LIST_LISTE_APP_MQGACF_AC	Paramètre de groupe de listes de distribution
ParameterCount	4	Nombre de structures de paramètres suivant la structure MQCFGR qui sont contenues dans ce groupe.

### **ObjectName**

Description : Nom d'une file d'attente dans la liste de distribution MQ\_Q\_NAME\_LENGTH. Inclus uniquement si des structures MQOR sont fournies.

- <sup>1</sup> Ce paramètre est inclus uniquement si l'objet ouvert est résolu en file d'attente et que la file d'attente est ouverte pour MQOO\_INPUT\_\*, MQOO\_OUTPUT ou MQOO\_BROWSE
- <sup>2</sup> Le paramètre QName ResolvedLocalest inclus uniquement s'il est différent du paramètre ResolvedQName .

Paramètre PCF: MQCACF\_NOM\_OBJET  
Niveau de trace : 2  
Type : MQCFST  
Longueur : MQ\_Q\_NAME\_LENGTH. Inclus uniquement si des structures MQOR sont fournies.

### **ObjectQMgrName**

Description : Nom du gestionnaire de files d'attente sur lequel la file d'attente nommée dans ObjectName est définie.  
Paramètre PCF: MQCACF\_OBJECT\_Q\_MGR\_NAME  
Niveau de trace : 2  
Type : MQCFST  
Longueur : MQ\_Q\_MGR\_NAME\_LENGTH. Inclus uniquement si des structures MQOR sont fournies.

### **CompCode**

Description : Code achèvement indiquant le résultat de l'ouverture de cet objet. Inclus uniquement si des structures MQRR sont fournies et que le code anomalie de MQOPEN est MQRC\_MULTIPLE\_MOTIFS  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 2  
Type : MQCFIN

### **Reason**

Description : Code anomalie indiquant le résultat de l'ouverture de cet objet. Inclus uniquement si des structures MQRR sont fournies et que le code anomalie de MQOPEN est MQRC\_MULTIPLE\_MOTIFS  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 2  
Type : MQCFIN

### **MQPUT**

L'application a démarré la fonction MQPUT MQI.

### **ObjectHandle**

Description : Descripteur d'objet  
Paramètre PCF: MQIACF\_HOBJ  
Niveau de trace : 1  
Type : MQCFIN

### **PutOptions**

Description : Options d'insertion de MQPMO.Options  
Paramètre PCF: OPTIONS MQIACF\_PUT\_MQ  
Niveau de trace : 1

Type : MQCFIN

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération

Paramètre PCF: CODE COMP\_MQIACF

Niveau de trace : 1

Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération

Paramètre PCF: MQIACF\_CODE\_RAISON

Niveau de trace : 1

Type : MQCFIN

### **MsgBuffer**

Description : Données de message.

Paramètre PCF: DONNÉES\_MESSAGE\_MQBACF\_MESSAGE

Niveau de trace : 1

Type : MQCFBS

Longueur : La longueur est régie par le paramètre TRACEDATA () défini dans la configuration APPTRACE. Si TRACEDATA=NONE, ce paramètre est omis.

### **MsgLength**

Description : Longueur du message.

Paramètre PCF: LONGUE\_MSG\_MQIACF\_MESSAGE

Niveau de trace : 1

Type : MQCFIN

### **RecsPresent**

Description : Nombre d'enregistrements de message d'insertion ou d'enregistrements de réponse présents. Inclus uniquement si MQPMO Version > = MQPMO\_VERSION\_2

Paramètre PCF: MQIACF\_RECS\_PRESENT

Niveau de trace : 1

Type : MQCFIN

### **KnownDestCount**

Description : Nombre de messages envoyés correctement aux files d'attente locales

Paramètre PCF: MQIACF\_KNOWN\_DEST\_COUNT

Niveau de trace : 1

Type : MQCFIN

### **UnknownDestCount**

Description : Nombre de messages envoyés correctement aux files d'attente éloignées  
Paramètre PCF: MQIACF\_UNKNOWN\_DEST\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

### **InvalidDestCount**

Description : Nombre de messages n'ayant pas pu être envoyés  
Paramètre PCF: MQIACF\_INVALID\_DEST\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

### **HighResTime**

Description : Heure de fonctionnement en microsecondes depuis minuit, le 1st 1970 (UTC)  
**Remarque :** La précision de cette minuterie varie en fonction de la prise en charge de la plateforme pour une minuterie à haute résolution.  
Paramètre PCF: MQIAMO64\_HIGHRES\_TIME  
Niveau de trace : 2  
Type : MQCFIN64

### **ObjectName**

Description : Nom de l'objet ouvert.  
Paramètre PCF: MQCACF\_NOM\_OBJET  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **ResolvedQName**

Description : Nom de la file d'attente après la résolution du nom de la file d'attente.  
Paramètre PCF: MQCACF\_RESOLVED\_Q\_NAME  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **ResolvedQMgrName**

Description : Nom du gestionnaire de files d'attente après la résolution de nom.  
Paramètre PCF: MQCACF\_RESOLVED\_Q\_MGR  
Niveau de trace : 2  
Type : MQCFST  
Longueur : MQ\_Q\_MGR\_NAME\_LENGTH

### **ResolvedLocalQName** <sup>3</sup>

Description : Contient le nom de la file d'attente locale une fois la résolution de nom effectuée.

Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME

Niveau de trace : 2

Type : MQCFST

### **ResolvedLocalQMgrName** <sup>3</sup>

Description : Contient le nom du gestionnaire de files d'attente local une fois la résolution de nom effectuée.

Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_MGR

Niveau de trace : 2

Type : MQCFST

Longueur : MQ\_Q\_MGR\_NAME\_LENGTH

### **ReportOptions**

Description : Options de rapport de message

Paramètre PCF: RAPPORT MQIACF\_RAPPORT

Niveau de trace : 2

Type : MQCFIN

### **MsgType**

Description : Type de message

Paramètre PCF: TYPE MQIACF\_MSG\_

Niveau de trace : 2

Type : MQCFIN

### **Expiry**

Description : Durée de vie des messages

Paramètre PCF: EXPIRATION\_MQIACF\_EXPIRATION

Niveau de trace : 2

Type : MQCFIN

### **Format**

Description : Nom de format des données de message

Paramètre PCF: MQCACH\_FORMAT\_NAME

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_FORMAT\_MQ

### **Priority**

Description : Priorité de message

Paramètre PCF: PRIO\_MQIACF\_ACF  
Niveau de trace : 2  
Type : MQCFIN

### ***Persistence***

Description : Persistence des messages  
Paramètre PCF: PERSISTENCE MQIACF\_PERSISTANCE  
Niveau de trace : 2  
Type : MQCFIN

### ***MsgId***

Description : Identificateur de message  
Paramètre PCF: ID\_MSG\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_MSG\_MQ\_ID

### ***CorrelId***

Description : Identificateur de corrélation  
Paramètre PCF: ID\_CORREL\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### ***ReplyToQueue***

Description :  
Paramètre PCF: MQCACF\_RÉPONSE\_À\_Q  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***ReplyToQMgr***

Description :  
Paramètre PCF: MQCACF\_REPLY\_TO\_Q\_MGR  
Niveau de trace : 2  
Type : MQCFST  
Longueur : MQ\_Q\_MGR\_NAME\_LENGTH

### ***CodedCharSetId***

Description : Identificateur de jeu de caractères des données de message  
Paramètre PCF: MQIA\_CODED\_CHAR\_SET\_ID

Niveau de trace : 2  
Type : MQCFIN

### **Encoding**

Description : Codage numérique des données de message.  
Paramètre PCF: CODAGE\_MQIACF\_CODAGE  
Niveau de trace : 2  
Type : MQCFIN

### **PutDate**

Description :  
Paramètre PCF: DATE MQCACF\_PUT\_DATE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_DATE\_PUT\_MQ\_

### **PutTime**

Description :  
Paramètre PCF: MQCACF\_HEURE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG

### **ResolvedQName**

Description : Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT\_Q.  
Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Niveau de trace : 2  
Type : MQCFST  
Longueur : MQ\_Q\_NAME\_LENGTH.

### **ResObjectString**

Description : Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT\_TOPIC.  
Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING  
Niveau de trace : 2  
Type : MQCFST  
Longueur : La longueur varie.

### **ResolvedType**

Description : Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT\_Q, MQOT\_TOPIC ou MQOT\_NONE.  
Paramètre PCF: MQIACF\_RESOLVED\_TYPE

Niveau de trace : 2  
Type : MQCFIN

### ***PolicyName***

Description : Nom de la règle qui a été appliquée à ce message.  
**Remarque :** Messages protégés AMS uniquement

Paramètre PCF: MQCA\_NOM\_REGLE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_OBJET\_MQ\_

### ***XmitqMsgId***

Description : ID du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQBACF\_XQH\_ID\_MSG  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_MSG\_MQ\_ID

### ***XmitqCorrelId***

Description : ID de corrélation du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQBACF\_XQH\_ID\_CORRÉLATION  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### ***XmitqPutTime***

Description : Heure d'insertion du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_HEURE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG

### ***XmitqPutDate***

Description : Date d'insertion du message dans l'en-tête de la file de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: DATE MQCACF\_XQH\_PUT\_DATE  
Niveau de trace : 2



Type : MQCFST  
Longueur : LONGUEUR\_DATE\_PUT\_MQ\_

### ***XmitqRemoteQName***

Description : Destination de file d'attente éloignée du message dans l'en-tête de la file d'attente de transmission.

**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_REMOTE\_Q\_NAME

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***XmitqRemoteQMgr***

Description : Destination du gestionnaire de files d'attente éloignées du message dans l'en-tête de la file d'attente de transmission.

**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_REMOTE\_Q\_MGR

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***PutMsgOptsStructure***

Description : Structure MQPMO.

Paramètre PCF: MQBACF\_MQPMO\_STRUCT

Niveau de trace : 3

Type : MQCFBS

Longueur : Longueur en octets de la structure MQPMO (la taille réelle dépend de la version de la structure)

### ***QMgrOpDuration***

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION

Niveau de trace : 2

Tapez MQCFIN64

---

<sup>3</sup> Le paramètre QName ResolvedLocalest inclus uniquement s'il est différent du paramètre ResolvedQName .

### *Structure d'en-tête de groupe PCF de la liste de distribution d'activité d'application MQPUT*

Si la fonction MQPUT place dans une liste de distribution, les paramètres MQPUT incluent un groupe PCF AppActivityDistList . Pour chacune des files d'attente de la liste de distribution, voir «Structure d'en-tête de groupe PCF de la liste de distribution d'activité d'application», à la page 265. Le groupe PCF AppActivityDistList combine les informations du MQPMR et les structures MQRR pour identifier les paramètres PUT et indiquer le résultat de l'opération PUT sur chaque file d'attente. Pour les opérations MQPUT, le groupe AppActivityDistList contient tout ou partie des paramètres suivants ( CompCode et Reason est présent si le code anomalie est MQRC\_MULTIPLE\_MOTIFS et que les autres paramètres sont déterminés par MQPMO.PutMsgRecFields ):

#### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération. Inclus uniquement si des structures MQRR sont fournies et que le code anomalie de MQPUT est MQRC\_MULTIPLE\_MOTIFS

Paramètre PCF: CODE\_COMP\_MQIACF

Niveau de trace : 2

Type : MQCFIN

#### **Reason**

Description : Code anomalie indiquant le résultat de l'insertion pour cet objet. Inclus uniquement si des structures MQRR sont fournies et que le code anomalie de MQPUT est MQRC\_MULTIPLE\_MOTIFS

Paramètre PCF: MQIACF\_CODE\_RAISON

Niveau de trace : 2

Type : MQCFIN

#### **MsgId**

Description : Identificateur de message. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_MSG\_ID

Paramètre PCF: ID\_MSG\_MQBACF

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_MSG\_MQ\_ID

#### **CorrelId**

Description : Identificateur de corrélation. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_CORREL\_ID

Paramètre PCF: ID\_CORREL\_MQBACF

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

#### **GroupId**

Description : Identificateur de groupe. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_GROUP\_ID

Paramètre PCF: ID\_GROUPE\_MQBACF

Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_GROUPE\_MQ\_ID

### **Feedback**

Description : Appréciations en Retour. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_FEEDBACK  
Paramètre PCF: MQIACF\_FEEDBACK  
Niveau de trace : 2  
Type : MQCFIN

### **AccountingToken**

Description : AccountingToken. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_ACCOUNTING\_TOKEN  
Paramètre PCF: JETON\_COMPTEUR\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : MQ\_ACCOUNTING\_TOKEN\_LENGTH.

### **MQPUT1**

L'application a démarré la fonction MQI MQPUT1

### **ObjectType**

Description : Type d'objet transmis dans MQOT.ObjectType  
Paramètre PCF: TYPE\_OBS\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **ObjectName**

Description : Nom de l'objet transmis à l'appel MQI avant toute tentative de résolution de nom de file d'attente.  
Paramètre PCF: MQCACF\_NOM\_OBJET  
Niveau de trace : 1  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **ObjectQMgrName**

Description : Nom du gestionnaire de files d'attente d'objet transmis à l'appel MQI avant toute tentative de résolution de nom de file d'attente.  
Paramètre PCF: MQCACF\_OBJECT\_Q\_MGR\_NAME  
Niveau de trace : 1  
Type : MQCFST  
Longueur : MQ\_Q\_MGR\_NAME\_LENGTH

**CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

**Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

**PutOptions**

Description : Options d'insertion de MQPMO.Options  
Paramètre PCF: OPTIONS MQIACF\_PUT\_MQ  
Niveau de trace : 1  
Type : MQCFIN

**AlternateUserId**

Description : Inclus uniquement si MQPMO\_ALTERNATE\_USER\_AUTHORITY est spécifié.  
Paramètre PCF: MQCACF\_ALTERNATE\_USERID  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_ID\_UTILISATEUR

**RecsPresent**

Description : Nombre d'enregistrements de nom d'objet présents  
Paramètre PCF: MQIACF\_RECS\_PRESENT  
Niveau de trace : 1  
Type : MQCFIN

**KnownDestCount**

Description : Nombre de files d'attente locales ouvertes correctement  
Paramètre PCF: MQIACF\_KNOWN\_DEST\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

**UnknownDestCount**

Description : Nombre de files d'attente éloignées ouvertes correctement  
Paramètre PCF: MQIACF\_UNKNOWN\_DEST\_COUNT  
Niveau de trace : 1

Type : MQCFIN

### ***InvalidDestCount***

Description : Nombre de files d'attente dont l'ouverture a échoué

Paramètre PCF: MQIACF\_INVALID\_DEST\_COUNT

Niveau de trace : 1

Type : MQCFIN

### ***MsgBuffer***

Description : Données de message.

Paramètre PCF: DONNÉES\_MESSAGE\_MQBACF\_MESSAGE

Niveau de trace : 1

Type : MQCFBS

Longueur : La longueur est régie par le paramètre TRACEDATA () défini dans la configuration APTRACE. Si TRACEDATA=NONE, ce paramètre est omis.

### ***MsgLength***

Description : Longueur du message.

Paramètre PCF: LONGUE\_MSG\_MQIACF\_MESSAGE

Niveau de trace : 1

Type : MQCFIN

### ***HighResTime***

Description : Heure de fonctionnement en microsecondes depuis minuit, le 1st 1970 (UTC)

**Remarque :** La précision de cette minuterie varie en fonction de la prise en charge par la plateforme d'une minuterie à haute résolution.

Paramètre PCF: MQIAMO64\_HIGHRES\_TIME

Niveau de trace : 2

Type : MQCFIN64

### ***ResolvedQName***

Description : Nom de la file d'attente après la résolution du nom de la file d'attente.

Paramètre PCF: MQCACF\_RESOLVED\_Q\_NAME

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***ResolvedQMgrName***

Description : Nom du gestionnaire de files d'attente après la résolution de nom.

Paramètre PCF: MQCACF\_RESOLVED\_Q\_MGR

Niveau de trace : 2

Type : MQCFST

Longueur : MQ\_Q\_MGR\_NAME\_LENGTH

#### ***ResolvedLocalQName*** <sup>4</sup>

Description : Contient le nom de la file d'attente locale une fois la résolution de nom effectuée

Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME

Niveau de trace : 2

Type : MQCFST

#### ***ResolvedLocalQMgrName*** <sup>4</sup>

Description : Contient le nom du gestionnaire de files d'attente local une fois la résolution de nom effectuée.

Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_MGR

Niveau de trace : 2

Type : MQCFST

Longueur : MQ\_Q\_MGR\_NAME\_LENGTH

#### ***AlternateSecurityId***

Description : Autre identificateur de sécurité. Présent uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_3 et MQOD.AlternateSecurityId n'est pas égal à MQSID\_NONE.

Paramètre PCF: MQBACF\_ALTERNATE\_SECURITYID

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_ID\_SÉCURITÉ\_MQ

#### ***ObjectString***

Description : Nom d'objet long. Inclus uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4 et à la zone VSLength de MQOD.ObjectString est MQVS\_NULL\_TERMINATED ou supérieure à zéro.

Paramètre PCF: MQCACF\_OBJECT\_CHAINE

Niveau de trace : 2

Type : MQCFST

Longueur : La longueur varie.

#### ***ResObjectString***

Description : Le nom d'objet long après le gestionnaire de files d'attente résout le nom fourni dans la zone ObjectName . Inclus uniquement pour les rubriques et les alias de file d'attente qui font référence à un objet de rubrique si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4 et VSLength est MQVS\_NULL\_TERMINATED ou supérieure à zéro.

Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING

Niveau de trace : 2

Type : MQCFST

Longueur : La longueur varie.

### ***ResolvedType***

Description : Type de l'objet résolu (de base) en cours d'ouverture. Inclus uniquement si MQOD.Version est supérieure ou égale à MQOD\_VERSION\_4. Les valeurs possibles sont MQOT\_Q, MQOT\_TOPIC ou MQOT\_NONE.

Paramètre PCF: MQIACF\_RESOLVED\_TYPE

Niveau de trace : 2

Type : MQCFIN

### ***ReportOptions***

Description : Options de rapport de message

Paramètre PCF: RAPPORT MQIACF\_RAPPORT

Niveau de trace : 2

Type : MQCFIN

### ***MsgType***

Description : Type de message

Paramètre PCF: TYPE MQIACF\_MSG\_

Niveau de trace : 2

Type : MQCFIN

### ***Expiry***

Description : Durée de vie des messages

Paramètre PCF: EXPIRATION\_MQIACF\_EXPIRATION

Niveau de trace : 2

Type : MQCFIN

### ***Format***

Description : Nom de format des données de message

Paramètre PCF: MQCACH\_FORMAT\_NAME

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_FORMAT\_MQ

### ***Priority***

Description : Priorité de message

Paramètre PCF: PRIO\_MQIACF\_ACF

Niveau de trace : 2

Type : MQCFIN

### **Persistence**

Description : Persistence des messages  
Paramètre PCF: PERSISTENCE MQIACF\_PERSISTANCE  
Niveau de trace : 2  
Type : MQCFIN

### **MsgId**

Description : Identificateur de message  
Paramètre PCF: ID\_MSG\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_MSG\_MQ\_ID

### **CorrelId**

Paramètre PCF: Identificateur de corrélation  
Description : ID\_CORREL\_MQBACF  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### **ReplyToQueue**

Description :  
Paramètre PCF: MQCACF\_RÉPONSE\_À\_Q  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **ReplyToQMgr**

Description :  
Paramètre PCF: MQCACF\_REPLY\_TO\_Q\_MGR  
Niveau de trace : 2  
Type : MQCFST  
Longueur : MQCFST

### **CodedCharSetId**

Description : Identificateur de jeu de caractères des données de message  
Paramètre PCF: MQIA\_CODED\_CHAR\_SET\_ID  
Niveau de trace : 2  
Type : MQCFIN



### **Encoding**

Description : Codage numérique des données de message.  
Paramètre PCF: CODAGE\_MQIACF\_CODAGE  
Niveau de trace : 2  
Type : MQCFIN

### **PutDate**

Description :  
Paramètre PCF: DATE MQCACF\_PUT\_DATE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_DATE\_PUT\_MQ\_

### **PutTime**

Description :  
Paramètre PCF: MQCACF\_HEURE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG

### **PolicyName**

Description : Nom de la règle qui a été appliquée à ce message.  
**Remarque :** Messages protégés AMS uniquement  
Paramètre PCF: MQCA\_NOM\_REGLE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_OBJET\_MQ\_

### **XmitqMsgId**

Description : ID du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER  
Paramètre PCF: MQBACF\_XQH\_ID\_MSG  
Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_MSG\_MQ\_ID

### **XmitqCorrelId**

Description : ID de corrélation du message dans l'en-tête de la file d'attente de transmission.  
**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER  
Paramètre PCF: MQBACF\_XQH\_ID\_CORRÉLATION

Niveau de trace : 2  
Type : MQCFBS  
Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### ***XmitqPutTime***

Description : Heure d'insertion du message dans l'en-tête de la file d'attente de transmission.

**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_HEURE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_TEMPS\_PUT\_MQ\_LONG

### ***XmitqPutDate***

Description : Date d'insertion du message dans l'en-tête de la file de transmission.

**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: DATE MQCACF\_XQH\_PUT\_DATE  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_DATE\_PUT\_MQ\_

### ***XmitqRemoteQName***

Description : Destination de file d'attente éloignée du message dans l'en-tête de la file d'attente de transmission.

**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_REMOTE\_Q\_NAME  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***XmitqRemoteQMGr***

Description : Destination du gestionnaire de files d'attente éloignées du message dans l'en-tête de la file d'attente de transmission.

**Remarque :** Uniquement lorsque le format est MQFMT\_XMIT\_Q\_HEADER

Paramètre PCF: MQCACF\_XQH\_REMOTE\_Q\_MGR  
Niveau de trace : 2  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### ***PutMsgOptsStructure***

Description : Structure MQPMO.  
Paramètre PCF: MQBACF\_MQPMO\_STRUCT

Niveau de trace : 3  
 Type : MQCFBS  
 Longueur : Longueur en octets de la structure MQPMO (la taille réelle dépend de la version de la structure)

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION

Niveau de trace : 2

Tapez MQCFIN64

### *MQPUT1 AppActivityDistList Structure d'en-tête de groupe PCF*

Si la fonction MQPUT1 est mise dans une liste de distribution, les paramètres de variable incluent un groupe PCF AppActivityDistList . Pour chacune des files d'attente de la liste de distribution, voir «Structure d'en-tête de groupe PCF de la liste de distribution d'activité d'application», à la page 265.

Le groupe PCF AppActivityDistList combine les informations des structures MQOR, MQPMR et MQRR pour identifier les objets et les paramètres PUT, et indique le résultat de l'opération PUT sur chaque file d'attente. Pour les opérations MQPUT1 , le groupe AppActivityDistList contient tout ou partie des paramètres suivants ( CompCode, Reason, ObjectNameet ObjectQMgrName) si le code anomalie est MQRC\_MULTIPLE\_MOTIFS et que les autres paramètres sont déterminés par MQPMO.PutMsgRecFields ):

### **CompCode**

Description : Code achèvement indiquant le résultat de l'insertion pour cet objet. Inclus uniquement si des structures MQRR sont fournies et que le code anomalie de MQPUT1 est MQRC\_MULTIPLE\_MOTIFS

Paramètre PCF: CODE COMP\_MQIACF

Niveau de trace : 2

Type : MQCFIN

### **Reason**

Description : Code anomalie indiquant le résultat de l'insertion pour cet objet. Inclus uniquement si des structures MQRR sont fournies et que le code anomalie de MQPUT1 est MQRC\_MULTIPLE\_MOTIFS

Paramètre PCF: MQIACF\_CODE\_RAISON

Niveau de trace : 2

Type : MQCFIN

<sup>4</sup> Le paramètre QName ResolvedLocalest inclus uniquement s'il est différent du paramètre ResolvedQName .

### **ObjectName**

Description : Nom d'une file d'attente dans la liste de distribution. Inclus uniquement si des structures MQOR sont fournies.

Paramètre PCF: MQCACF\_NOM\_OBJET

Niveau de trace : 2

Type : MQCFST

Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **MsgId**

Description : Identificateur de message. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_MSG\_ID

Paramètre PCF: ID\_MSG\_MQBACF

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_MSG\_MQ\_ID

### **CorrelId**

Description : Identificateur de corrélation. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_CORREL\_ID

Paramètre PCF: ID\_CORREL\_MQBACF

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_ID\_CORREL\_MQ\_

### **GroupId**

Description : Identificateur de groupe. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_GROUP\_ID

Paramètre PCF: ID\_GROUPE\_MQBACF

Niveau de trace : 2

Type : MQCFBS

Longueur : LONGUEUR\_GROUPE\_MQ\_ID

### **Feedback**

Description : Appréciations en Retour. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_FEEDBACK

Paramètre PCF: MQIACF\_FEEDBACK

Niveau de trace : 2

Type : MQCFIN

### **AccountingToken**

Description : AccountingToken. Inclus uniquement si les structures MQPMR sont provided.and PutMsgRecFields inclut MQPMRF\_ACCOUNTING\_TOKEN

Paramètre PCF: JETON\_COMPTEUR\_MQBACF

Niveau de trace : 2  
Type : MQCFBS  
Longueur : MQ\_ACCOUNTING\_TOKEN\_LENGTH.

### *MQSET*

L'application a démarré la fonction MQSET MQI

#### **ObjectHandle**

Description : Descripteur d'objet  
Paramètre PCF: MQIACF\_HOBJ  
Niveau de trace : 1  
Type : MQCFIN

#### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

#### **Reason**

Description : Résultat du code anomalie de l'opération  
Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

#### **SelectorCount**

Description : Nombre de sélecteurs fournis dans le tableau Sélecteurs.  
Paramètre PCF: MQIACF\_SELECTOR\_COUNT  
Niveau de trace : 2  
Type : MQCFIN

#### **Selectors**

Description : Liste des attributs (entiers ou caractères) dont les valeurs sont mises à jour par MQSET.  
Paramètre PCF: Sélecteurs MQIACF\_SELECTORS  
Niveau de trace : 2  
Type : MQCFIL

#### **ResolvedQName**

Description : Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT\_Q.  
Paramètre PCF: MQCACF\_RESOLVED\_LOCAL\_Q\_NAME  
Niveau de trace : 2

Tapez MQCFST  
Longueur : MQ\_Q\_NAME\_LENGTH.

### ***ResObjectString***

Description : Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT\_TOPIC.  
Paramètre PCF: MQCACF\_RESOLVED\_OBJECT\_STRING  
Niveau de trace : 2  
Tapez MQCFST  
Longueur : La longueur varie.

### ***ResolvedType***

Description : Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT\_Q, MQOT\_TOPIC ou MQOT\_NONE.  
Paramètre PCF: MQIACF\_RESOLVED\_TYPE  
Niveau de trace : 2  
Tapez MQCFIN

### ***IntAttrCount***

Description : Nombre d'attributs de type entier à mettre à jour par l'opération d'ensemble.  
Paramètre PCF: MQIACF\_INTATTR\_COUNT  
Niveau de trace : 3  
Type : MQCFIN

### ***IntAttrrs***

Description : Valeurs d'attribut entières  
Paramètre PCF: MQIACF\_INT\_ATTRS  
Niveau de trace : 3  
Type : MQCFIL  
Plage : Ce paramètre n'est présent que si le nombre IntAttrrest > 0

### ***CharAttrrs***

Description : Attributs de caractères à mettre à jour par l'opération de définition. Les valeurs sont concaténées.  
Paramètre PCF: MQCACF\_CHAR\_ATTRS  
Niveau de trace : 3  
Type : MQCFST  
Plage : Ce paramètre est inclus uniquement si la longueur CharAttrrest > 0

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION

Niveau de trace : 2

Tapez MQCFIN64

### **MQSUB**

L'application a démarré la fonction MQI MQSUB

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération

Paramètre PCF: CODE COMP\_MQIACF

Niveau de trace : 1

Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération

Paramètre PCF: MQIACF\_CODE\_RAISON

Niveau de trace : 1

Type : MQCFIN

### **SubHandle**

Description : Le descripteur d'abonnement

Paramètre PCF: MQIACF\_HSUB

Niveau de trace : 1

Type : MQCFIN

### **ObjectHandle**

Description : Descripteur d'objet

Paramètre PCF: MQIACF\_HOBJ

Niveau de trace : 1

Type : MQCFIN

### **Options**

Description : Options d'abonnement

Paramètre PCF: MQIACF\_SUB\_OPTIONS

Niveau de trace : 1

Type : MQCFIN

### **ObjectName**

Description : le nom de l'objet,  
Paramètre PCF: MQCACF\_NOM\_OBJET  
Niveau de trace : 1  
Type : MQCFST  
Longueur : LONGUEUR\_NOM\_Q\_MQ\_

### **ObjectString**

Description : Nom d'objet long.  
Paramètre PCF: MQCACF\_OBJECT\_CHAINE  
Niveau de trace : 1  
Type : MQCFST  
Plage : Inclus uniquement si la zone VSLength de MQSD.ObjectString est supérieur à zéro ou MQVS\_NULL\_TERMINATED.  
Longueur : La longueur varie.

### **AlternateUserId**

Description :  
Paramètre PCF: MQCACF\_ALTERNATE\_USERID  
Niveau de trace : 2  
Type : MQCFST  
Plage : Inclus uniquement si MQSO\_ALTERNATE\_USER\_AUTHORITY est spécifié.  
Longueur : LONGUEUR\_ID\_UTILISATEUR

### **AlternateSecurityId**

Description : Autre identificateur de sécurité.  
Paramètre PCF: MQBACF\_ALTERNATE\_SECURITYID  
Niveau de trace : 2  
Type : MQCFBS  
Plage : Présente uniquement si MQSO\_ALTERNATE\_USER\_AUTHORITY est spécifié et MQSD.AlternateSecurityId n'est pas égal à MQSID\_NONE.  
Longueur : LONGUEUR\_ID\_SÉCURITÉ\_MQ

### **SubName**

Description : Nom abonnement  
Paramètre PCF: MQCACF\_NOM\_SOUS-SYSTÈME  
Niveau de trace : 2  
Type : MQCFST  
Plage : Inclus uniquement si la zone VSLength de MQSD.SubName est supérieure à zéro ou si MQVS\_NULL\_TERMINATED.  
Longueur : La longueur varie.



### ***SubUserData***

Description :	Données utilisateur d'abonnement
Paramètre PCF:	DONNEES_SOUS-UTILISATEUR_MQCACF
Niveau de trace :	2
Type :	MQCFST
Plage :	Inclus uniquement si la zone VSLength de MQSD.SubName est supérieure à zéro ou si MQVS_NULL_TERMINATED.
Longueur :	La longueur varie.

### ***SubCorrelId***

Description :	Identificateur de corrélation d'abonnement
Paramètre PCF:	ID_SUB_CORRÉLATION_MQBACF
Niveau de trace :	2
Type :	MQCFBS
Longueur :	LONGUEUR_ID_CORREL_MQ_

### ***SelectionString***

Description :	Chaîne de sélection.
Paramètre PCF:	CHAÎNE_SÉLECTION_MQCACF
Niveau de trace :	2
Type :	MQCFST
Plage :	Inclus uniquement si la zone VSLength de l'instruction MQSD est définie. SelectionString est MQVS_NULL_TERMINATED ou supérieur à zéro.
Longueur :	La longueur varie.

### ***ResolvedQName***

Description :	Le nom de la file d'attente désigné par ObjectHandle, lorsque ResolvedType est MQOT_Q.
Paramètre PCF:	MQCACF_RESOLVED_LOCAL_Q_NAME
Niveau de trace :	2
Tapez	MQCFST
Longueur :	MQ_Q_NAME_LENGTH.

### ***ResObjectString***

Description :	Nom d'objet désigné par ObjectHandle, lorsque ResolvedType est MQOT_TOPIC.
Paramètre PCF:	MQCACF_RESOLVED_OBJECT_STRING
Niveau de trace :	2
Tapez	MQCFST
Longueur :	La longueur varie.

### **ResolvedType**

Description :	Type de l'objet référencé par l'objet ObjectHandle. Les valeurs possibles sont MQOT_Q, MQOT_TOPIC ou MQOT_NONE.
Paramètre PCF:	MQIACF_RESOLVED_TYPE
Niveau de trace :	2
Tapez	MQCFIN

### **SubDescriptorStructure**

Description :	Structure MQSD.
Paramètre PCF:	MQBACF_MQSD_STRUCT
Niveau de trace :	3
Type :	MQCFBS
Longueur :	Longueur en octets de la structure MQSD.

### **QMGrOpDuration**

Description :	Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  <b>Remarque :</b> La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.
Paramètre PCF:	MQIAMO64_QMGR_OP_DURATION
Niveau de trace :	2
Tapez	MQCFIN64

### **MQSUBRQ**

L'application a démarré la fonction MQSUBRQ MQI

### **CompCode**

Description :	Code achèvement indiquant le résultat de l'opération
Paramètre PCF:	CODE COMP_MQIACF
Niveau de trace :	1
Type :	MQCFIN

### **Reason**

Description :	Résultat du code anomalie de l'opération
Paramètre PCF:	MQIACF_CODE_RAISON
Niveau de trace :	1
Type :	MQCFIN

### **SubHandle**

Description :	Le descripteur d'abonnement
Paramètre PCF:	MQIACF_HSUB

Niveau de trace : 1  
Type : MQCFIN

### **SubOptions**

Description : Sous-options de MQSB.Options  
Paramètre PCF: OPTIONS MQIACF\_SUBRQ\_MQ  
Niveau de trace : 2  
Type : MQCFIN

### **Action**

Description : Action de demande d'abonnement (MQSR\_\*)  
Paramètre PCF: MQIACF\_SUBRQ\_ACTION  
Niveau de trace : 2  
Type : MQCFIN

### **NumPubs**

Description : Nombre de publications envoyées à la suite de cet appel (à partir de MQSB.NumPubs)  
Paramètre PCF: MQIACF\_NUM\_PUBS  
Niveau de trace : 2  
Type : MQCFIN

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.  
  
La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .  
**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.  
  
Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez MQCFIN64

### **MQSTAT**

L'application a démarré la fonction MQI MQSTAT

### **CompCode**

Description : Code achèvement indiquant le résultat de l'opération  
Paramètre PCF: CODE COMP\_MQIACF  
Niveau de trace : 1  
Type : MQCFIN

### **Reason**

Description : Résultat du code anomalie de l'opération

Paramètre PCF: MQIACF\_CODE\_RAISON  
Niveau de trace : 1  
Type : MQCFIN

### **Type**

Description : Type d'informations de statut demandées  
Paramètre PCF: TYPE\_STATUS\_MQIACF  
Niveau de trace : 2  
Type : MQCFIN

### **StatusStructure**

Description : Structure MQSTS.  
Paramètre PCF: MQBACF\_MQSTS\_STRUCT  
Niveau de trace : 3  
Type : MQCFBS  
Longueur : Longueur en octets de la structure MQSTS (la taille réelle dépend de la version de la structure)

### **QMGrOpDuration**

Description : Durée approximative des appels API, en microsecondes, dans le gestionnaire de files d'attente.

La durée n'inclut pas le temps passé en dehors du gestionnaire de files d'attente. Par exemple, la durée utilisée en tant que client IBM MQ .

**Remarque :** La précision de ce minuteur varie en fonction de la plateforme utilisée par votre entreprise.

Paramètre PCF: MQIAMO64\_QMGR\_OP\_DURATION  
Niveau de trace : 2  
Tapez : MQCFIN64

### **Paramètres de variable pour les opérations XA d'activité d'application**

Les opérations XA sont des appels d'API que les applications peuvent effectuer pour permettre à MQ de participer à une transaction. Les paramètres de chaque opération sont définis dans la section suivante.

Le niveau de trace indique le niveau de granularité de trace requis pour que les paramètres soient inclus dans la trace. Les valeurs de niveau de trace possibles sont les suivantes:

#### 1. Bas

Le paramètre est inclus lorsque le traçage d'activité "faible", "moyenne" ou "élevée" est configuré pour une application. Ce paramètre signifie qu'un paramètre est toujours inclus dans le groupe AppActivityData pour l'opération. Cet ensemble de paramètres est suffisant pour tracer les appels MQI émis par une application et pour voir s'ils ont abouti.

#### 2. Moyen

Le paramètre est uniquement inclus dans le groupe AppActivityData pour l'opération lorsque la fonction de trace de l'activité "moyenne" ou "élevée" est configurée pour une application. Cet ensemble de paramètres ajoute des informations sur les ressources, par exemple, les noms de file d'attente et de rubrique utilisés par l'application.

#### 3. Elevé

Le paramètre est uniquement inclus dans le groupe AppActivityData pour l'opération lorsque le traçage d'activité "élevé" est configuré pour une application. Cet ensemble de paramètres inclut des vidages mémoire des structures transmises aux fonctions MQI et XA. Pour cette raison, il contient des informations supplémentaires sur les paramètres utilisés dans les appels MQI et XA. Les vidages mémoire de la structure sont des copies peu profondes des structures. Pour éviter les tentatives erronées de déréférencement des pointeurs, les valeurs de pointeur dans les structures sont définies sur NULL.

**Remarque :** La version de la structure faisant l'objet d'un vidage n'est pas nécessairement identique à la version utilisée par une application. La structure peut être modifiée par un exit de croisement d'API, par le code de trace d'activité ou par le gestionnaire de files d'attente. Un gestionnaire de files d'attente peut modifier une structure vers une version ultérieure, mais le gestionnaire de files d'attente ne la remplace jamais par une version antérieure de la structure. Pour ce faire, vous risquez de perdre des données.

#### *AXREG*

L'application a démarré la fonction AXREG AX

#### ***XID***

Description :	Structure XID
Paramètre PCF:	MQBACF_XA_XID
Niveau de trace :	1
Type :	MQCFBS
Longueur :	Taille (XID)

#### ***Rmid***

Description :	Identificateur du gestionnaire de ressources
Paramètre PCF:	MQIACF_XA_RMID
Niveau de trace :	1
Type :	MQCFIN

#### ***Flags***

Description :	Indicateurs
Paramètre PCF:	MQIACF_XA_FLAGS
Niveau de trace :	1
Type :	MQCFIN

#### ***XARetCode***

Description :	Code retour
Paramètre PCF:	MQIACF_XA_RETCODE
Niveau de trace :	1
Type :	MQCFIN

#### *AXUNREG*

L'application a démarré la fonction AXUNREG AX

#### ***Rmid***

Description :	Identificateur du gestionnaire de ressources
---------------	--

Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

### **Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

### **XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

### **XACLOSE**

L'application a démarré la fonction XACLOSE AX

### **Xa\_info**

Description : Informations utilisées pour initialiser le gestionnaire de ressources.  
Paramètre PCF: MQCACF\_XA\_INFO  
Niveau de trace : 1  
Type : MQCFST

### **Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

### **Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

### **XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

## *XACOMMIT*

L'application a démarré la fonction XACOMMIT AX

### ***XID***

Description : Structure XID  
Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

### ***Rmid***

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

### ***Flags***

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

### ***XARetCode***

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

## *XATERMINE*

L'application a démarré la fonction XACOMplete AX

### ***Handle***

Description : Traitement de l'opération asynchrone  
Paramètre PCF: MQIACF\_XA\_HANDLE  
Niveau de trace : 1  
Type : MQCFIN

### ***Retval***

Description : Valeur de retour de la fonction asynchrone  
Paramètre PCF: MQIACF\_XA\_RETVAL  
Niveau de trace : 1  
Type : MQCFINMQCFBS

**Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

**Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

**XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

**XAEND**

L'application a démarré la fonction XAEND AX

**XID**

Description : Structure XID  
Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

**Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

**Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

**XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE



Niveau de trace : 1  
Type : MQCFIN

#### *XAFORGET*

L'application a démarré la fonction AXREG AX

#### **XID**

Description : Structure XID  
Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

#### **Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

#### **Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

#### **XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

#### *XAOPEN*

L'application a démarré la fonction XAOPEN AX

#### **Xa\_info**

Description : Informations utilisées pour initialiser le gestionnaire de ressources.  
Paramètre PCF: MQCACF\_XA\_INFO  
Niveau de trace : 1  
Type : MQCFST

#### **Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1

Type : MQCFIN

### **Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

### **XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

### **XAPREPARE**

L'application a démarré la fonction XAPREPARE AX

### **XID**

Description : Structure XID  
Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

### **Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

### **Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

### **XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

### **XARECOVER**

L'application a démarré la fonction XARECOVER AX

### **Count**

Description : Nombre de XID  
Paramètre PCF: MQIACF\_XA\_COUNT  
Niveau de trace : 1  
Type : MQCFIN

### **XIDs**

Description : Les structures XID

**Remarque :** Il existe plusieurs instances de ce paramètre PCF-une pour chaque structure XID jusqu'à compter les XID

Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

### **Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

### **Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

### **XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

### **XAROLLBACK**

L'application a démarré la fonction XAROLLBACK AX

### **XID**

Description : Structure XID  
Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

**Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

**Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

**XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE  
Niveau de trace : 1  
Type : MQCFIN

**XASTART**

L'application a démarré la fonction XASTART AX

**XID**

Description : Structure XID  
Paramètre PCF: MQBACF\_XA\_XID  
Niveau de trace : 1  
Type : MQCFBS  
Longueur : Taille (XID)

**Rmid**

Description : Identificateur du gestionnaire de ressources  
Paramètre PCF: MQIACF\_XA\_RMID  
Niveau de trace : 1  
Type : MQCFIN

**Flags**

Description : Indicateurs  
Paramètre PCF: MQIACF\_XA\_FLAGS  
Niveau de trace : 1  
Type : MQCFIN

**XARetCode**

Description : Code retour  
Paramètre PCF: MQIACF\_XA\_RETCODE

Niveau de trace : 1  
Type : MQCFIN

## Multi Rubriques système pour la surveillance et la trace de l'activité

Les rubriques système dans les arborescences de rubriques du gestionnaire de files d'attente sont utilisées pour la surveillance des ressources (dont certaines sont similaires au contenu des messages de statistiques) et comme moyen de consommer la trace de l'activité de l'application.

### Branche \$SYS/MQ de l'arborescence de rubriques du gestionnaire de files d'attente

L'arborescence de rubriques de chaque gestionnaire de files d'attente contient la branche \$SYS/MQ . Le gestionnaire de files d'attente publie des données dans des chaînes de rubrique dans cette branche. Un utilisateur autorisé peut s'abonner à ces chaînes de rubrique afin de recevoir des informations sur le gestionnaire de files d'attente et sur l'activité ayant lieu sur ce gestionnaire de files d'attente. Ces rubriques système sont utilisées pour la trace et la surveillance de l'activité des applications. Pour plus d'informations, voir [Arborescences de rubriques](#).

La racine de la branche \$SYS/MQ est représentée par l'objet de rubrique SYSTEM.ADMIN.TOPIC. La branche \$SYS/MQ de l'arborescence de rubriques est isolée du reste de l'arborescence de rubriques comme suit :

- Un abonnement souscrit avec des caractères génériques à un niveau dans l'arborescence supérieur à \$SYS/MQ ne correspond à aucune chaîne de rubrique dans la branche \$SYS/MQ. L'opération avec caractères génériques pour SYSTEM.ADMIN.TOPIC a pour valeur "Block" et ne peut pas être modifiée. Cette limitation s'applique également lorsque vous utilisez des caractères génériques avec la commande **runmqsc DISPLAY TPSTATUS** afin d'afficher des noeuds dans l'arborescence de rubriques. Pour afficher les noeuds de rubrique dans la branche \$SYS/MQ, commencez la chaîne de rubrique par \$SYS/MQ. Par exemple, utilisez \$SYS/MQ/# pour afficher tous les noeuds.
- Vous devez être autorisé à utiliser l'arborescence de rubriques \$SYS/MQ ou à une valeur supérieure à \$SYS/MQ pour pouvoir utiliser cette arborescence. L'autorisation pour l'abonnement à une chaîne de rubriques s'appuie sur l'autorisation accordée pour un objet de rubrique administré au niveau de la chaîne de rubrique ou à un niveau supérieur dans l'arborescence de rubriques. Les autorisations qui sont accordées à la racine (SYSTEM.BASE.TOPIC) permettent à un utilisateur d'accéder à toutes les chaînes de rubrique. Toutefois, dans le cas de la branche \$SYS/MQ, l'accès accordé à un niveau supérieur à \$SYS/MQ ne s'applique pas aux chaînes de rubrique \$SYS/MQ.
- La branche \$SYS/MQ de l'arborescence de rubriques est isolée des attributs de rubrique qui sont définis à un niveau supérieur dans l'arborescence. SYSTEM.ADMIN.TOPIC n'hérite pas des attributs d'un objet de rubrique défini à un niveau supérieur dans l'arborescence de rubriques. Par exemple, la modification des attributs de SYSTEM.BASE.TOPIC n'a pas d'impact sur le comportement de la branche \$SYS/MQ.

Toutes les chaînes de rubrique qui commencent par \$SYS/MQ sont réservées pour IBM MQ. Elles présentent les restrictions suivantes :

- Vous ne pouvez pas activer la multidiffusion depuis la branche \$SYS/MQ de l'arborescence de rubriques.
- La mise en cluster n'est pas prise en charge pour la branche \$SYS/MQ.
- Le mécanisme d'abonnement de proxy ne peut pas avoir pour valeur "force".
- Les applications ne peuvent pas publier de données dans une chaîne de rubrique \$SYS/MQ.
- La portée de la publication et de l'abonnement est par défaut le gestionnaire de files d'attente local seulement.
- Aucun caractère générique ne peut être utilisé aux niveaux suivants :
  - \$SYS/MQ/
  - \$SYS/MQ/INFO
  - \$SYS/MQ/INFO/QMGR

- \$SYS/MQ/INFO/QMGR/nom\_gest\_files\_attente
- \$SYS/MQ/INFO/QMGR/nom\_gest\_files\_attente/ActivityTrace

Si vous tentez d'utiliser des caractères génériques à ces niveaux, l'abonnement échoue avec le motif MQRC\_ADMIN\_TOPIC\_STRING\_ERROR.

## API de surveillance des performances disponible via la publication / l'abonnement

Vous pouvez surveiller les statistiques du gestionnaire de files d'attente à l'aide du mécanisme de publication / abonnement simple et du modèle d'application amqsrua ou de votre propre application. Les statistiques sont publiées dans une rubrique système sous \$SYS/MQ/INFO/QMGR qui aide les utilisateurs à surveiller les ressources. Ces statistiques peuvent être visualisées en exécutant l'exemple d'application amqsrua ou en écrivant une application qui s'abonne à la rubrique système de surveillance des ressources de la même manière que **amqsrua**.

### Concepts associés

«Trace de l'activité d'application», à la page 212

La trace de l'activité des applications produit des informations détaillées sur le comportement des applications connectées à un gestionnaire de files d'attente. Il trace le comportement d'une application et fournit une vue détaillée des paramètres utilisés par une application lors de son interaction avec les ressources IBM MQ . Elle affiche également la séquence des appels MQI émis par une application.

## Windows Linux Surveillance de l'utilisation des ressources système à l'aide de la commande amqsrua

Vous pouvez utiliser la commande **amqsrua** pour interroger les données de performances liées à l'utilisation des ressources système d'un gestionnaire de files d'attente.

### Pourquoi et quand exécuter cette tâche

L'exemple d'application amqsrua présente un moyen de consommer des publications de surveillance IBM MQ et d'afficher des données de performances publiées par les gestionnaires de files d'attente. Ces données peuvent inclure des informations sur l'utilisation de l'unité centrale, de la mémoire et du disque. Vous pouvez aussi afficher des données équivalentes aux données statistiques de format de commande programmable (PCF) STATMQI. Les données sont publiées toutes les dix secondes et affichées alors que la commande s'exécute.

Vous pouvez exécuter la commande uniquement avec le nom du gestionnaire de files d'attente et en mode interactif pour choisir les paramètres **CLASS**, **TYPE** puis **object** disponibles pour le gestionnaire de files d'attente à chaque étape. Si vous connaissez les noms **CLASS**, **TYPE** et **object** pour lesquels vous souhaitez afficher des informations, vous pouvez les spécifier lorsque vous exécutez la commande **amqsrua** .

Par défaut, l'application amqsrua recherche les statistiques qui sont publiées par le gestionnaire de files d'attente sous l'arborescence de rubriques \$SYS/MQ/INFO/QMGR. D'autres composants ou applications peuvent utiliser un mécanisme similaire pour publier sous un point de départ de rubrique différent. Depuis la IBM MQ 9.1.0, vous pouvez utiliser le paramètre **-p** pour indiquer où amqsrua recherche les statistiques pour ces autres composants sur Linux et Windows.

#### -m

Nom du gestionnaire de files d'attente Il doit être en cours d'exécution. Si vous ne spécifiez pas de nom de gestionnaire de files d'attente, les statistiques du gestionnaire de files d'attente par défaut sont affichées.

#### -c

Nom de la classe. Les publications d'utilisation des ressources IBM MQ sont associées à une classe. Les classes représentent le niveau supérieur dans l'arborescence des métadonnées qui décrit les informations d'utilisation des ressources disponibles.

#### UC

Renvoie des informations sur l'utilisation de l'unité centrale.

## DISQUE

Renvoie des informations sur l'utilisation du disque.

## STATMQI

Renvoie des informations sur l'utilisation MQI.

## STATQ

Renvoie des informations sur l'utilisation MQI par file d'attente.

## STATAPP

Renvoie des informations sur les statistiques d'utilisation de l'application spécifiée.

Voir «[Développement de votre propre application de surveillance des ressources](#)», à la page 305 pour plus d'informations sur la façon de spécifier l'arborescence de rubriques pour les métadonnées et [Utilisation du nom d'application dans les langages de programmation pris en charge](#) pour les caractères que vous pouvez utiliser dans les noms d'application.

**Remarque :** L'option **-o** permet d'indiquer le nom de l'application surveillée pour STATAPP et le nom de la file d'attente pour STATQ.

### -t

Nom TYPE. Les publications d'utilisation des ressources IBM MQ sont associées à un type au sein d'une classe. Chaque publication inclut la classe et le type qui permettent de trouver les définitions `class/type/element` et de traiter les publications résultantes. Les descriptions `class/type/element` sont publiées en tant que métadonnées au démarrage du gestionnaire de files d'attente.

### -o

Nom de l'objet. Les publications d'utilisation des ressources sont des messages PCF qui consistent en une séquence d'éléments PCF. Les éléments PCF qui sont publiés pour chaque paire classe / type sont annoncés dans les métadonnées. Une feuille est stockée dans l'arbre qui décrit chaque élément, ce qui permet de traiter les éléments.

### -p

Préfixe de métadonnées. Spécifiez un point de départ d'arborescence de rubriques où **amqsrua** peut rechercher les statistiques publiées par les gestionnaires de files d'attente. L'arborescence de rubriques par défaut est `$$SYS/MQ/INFO/QMGR`, mais d'autres composants ou applications peuvent publier des statistiques sous un autre point de départ d'arborescence de rubriques.

### -n

Nombre de publications. Vous pouvez spécifier le nombre de rapports renvoyés avant la fin de la commande. Les données sont publiées environ toutes les dix secondes. Par conséquent, si vous entrez la valeur 50, la commande renvoie 50 rapports de plus de 500 secondes. Si vous ne spécifiez pas ce paramètre, la commande s'exécute jusqu'à ce qu'une erreur survienne ou que le gestionnaire de files d'attente s'arrête.

### -s

File d'attente modèle. Indiquez éventuellement une file d'attente modèle à utiliser. (Par défaut, **amqsrua** utilise `SYSTEM.DEFAULT.MODEL.QUEUE`.)

### -h

Utilisation

## Procédure

1. Dans le répertoire des exemples, exécutez la commande suivante pour afficher les données disponibles pour le gestionnaire de files d'attente:

-  Sous Linux, `MQ_INSTALLATION_PATH/samp/bin`:

```
./amqsrua -m QMgrName
```

-  Sous Windows, `MQ_INSTALLATION_PATH\tools\c\Samples\Bin64`:

```
amqsrua -m QMgrName
```

où *QMgrName* indique le nom du gestionnaire de files d'attente que vous souhaitez interroger. Il doit être en cours d'exécution. Si vous ne spécifiez pas de nom de gestionnaire de files d'attente, le gestionnaire de files d'attente par défaut est utilisé.

Les options suivantes sont disponibles :

```
CPU : Platform central processing units
DISK : Platform persistent data stores
STATMQI : API usage statistics
STATQ : API per-queue usage statistics
Enter Class selection
==>
```

2. Dans la liste des options CLASS, entrez STATMQI.

```
==> STATMQI
CONNDISC : MQCONN and MQDISC
OPENCLOSE : MQOPEN and MQCLOSE
INQSET : MQINQ and MQSET
PUT : MQPUT
GET : MQGET
SYNCPOINT : Commit and rollback
SUBSCRIBE : Subscribe
PUBLISH : Publish
Enter Type selection
==>
```

3. Dans la liste des options TYPE, entrez PUT.

```
==>PUT
Publication received PutDate:20170329 PutTime:17045485 Interval:4 minutes,13.978 seconds
Interval total MQPUT/MQPUT1 count 22
Interval total MQPUT/MQPUT1 byte count 25284 100/sec
Non-persistent message MQPUT count 22
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 25284 100/sec
Put persistent messages - byte count 0
MQSTAT count 0

Publication received PutDate:20170329 PutTime:17050485 Interval:10.001 seconds
Interval total MQPUT/MQPUT1 count 1
Interval total MQPUT/MQPUT1 byte count 524 52/sec
Non-persistent message MQPUT count 1
Persistent message MQPUT count 0
Failed MQPUT count 0
Non-persistent message MQPUT1 count 0
Persistent message MQPUT1 count 0
Failed MQPUT1 count 0
Put non-persistent messages - byte count 524 52/sec
Put persistent messages - byte count 0
MQSTAT count 0
```

## Résultats

Vous avez utilisé le modèle d'application **amqszrua** de manière interactive pour examiner les statistiques que les gestionnaires de files d'attente publient sur les rubriques système sous le préfixe de métadonnées \$SYS/MQ/INFO/QMGR.

**Remarque :** Les classes et les types de publications de ressources disponibles pour les gestionnaires de files d'attente peuvent varier en fonction de leur configuration, de leur version et de leur plateforme. Utilisez **amqszrua** de manière interactive pour rechercher les classes, les types et les éléments disponibles pour votre gestionnaire de files d'attente spécifique.

## Que faire ensuite

Pour développer votre propre application de surveillance, voir [«Développement de votre propre application de surveillance des ressources»](#), à la page 305.



## Tâches associées

[Préparation et exécution d'exemples de programmes sous Windows](#)

[Préparation et exécution d'exemples de programmes sous UNIX et Linux](#)

## **ALW** Développement de votre propre application de surveillance des ressources

Vous pouvez développer votre propre application pour surveiller les ressources système.

Chaque gestionnaire de files d'attente publie des données d'utilisation des ressources dans des rubriques. Ces données sont utilisées par les abonnés à ces rubriques. Lorsqu'un gestionnaire de files d'attente démarre, il publie un ensemble de messages dans des métarubriques. Ces messages indiquent quelles rubriques d'utilisation des ressources sont prises en charge par le gestionnaire de files d'attente et décrivent le contenu des messages publiés dans ces rubriques. Les outils d'administration peuvent s'abonner aux métadonnées pour découvrir quelles informations d'utilisation des ressources sont disponibles et sur quelles rubriques, puis s'abonner aux rubriques annoncées.

L'arborescence de rubriques des métadonnées possède la structure suivante:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/instance]/type]
```

Utilisez la structure suivante pour les métadonnées de l'arborescence de rubriques:

```
$SYS/MQ/INFO/QMGR/QMGR-NAME/Monitor/class[/resourceid]/type]
```

pour préciser que vous spécifiez la ressource surveillée, dont le nom d'application ne doit pas dépasser 28 caractères, et pour éviter toute confusion avec les attributs de type de métadonnées internes.

Pour obtenir la liste des classes possibles, voir «[Surveillance de l'utilisation des ressources système à l'aide de la commande amqsrua](#)», à la page 302.

Le code source de l'application amqsrua est fourni en tant qu'exemple IBM MQ . Vous pouvez utiliser cet exemple d'application comme guide pour la création de votre propre application de surveillance. Vous pouvez extraire la source de l'exemple depuis une installation de client IBM MQ. Le fichier source est appelé amqsrua.c et se trouve dans le répertoire des exemples :

- **Linux** **AIX** Sur les plateformes AIX and Linux , `MQ_INSTALLATION_PATH/samp/`
- **Windows** Sur les plateformes Windows , `MQ_INSTALLATION_PATH\tools\c\Samples\`

L'application amqsrua s'abonne aux rubriques d'utilisation des ressources IBM MQ et formate les données PCF publiées qui en résultent. La source de l'application fournit un exemple de base expliquant comment s'abonner à et consommer ce type de données d'administration. L'application amqsrua effectue les tâches suivantes:



- Il crée un abonnement non durable aux rubriques identifiées par les paramètres d'entrée.
- Il appelle MQGET régulièrement pour obtenir des messages des rubriques et écrire des données dans stdout.
- Il écrit un message pour chaque motif MQI (autre que MQRC\_NONE).
- Il s'arrête si le code d'achèvement MQI est MQCC\_FAILED ou lorsque le nombre de publications d'utilisation des ressources demandé a été consommé.

## **Multi** Métriques publiées sur les rubriques système

Les indicateurs sont catégorisés en classes et sous-catégorisés en types. Plusieurs métriques sont publiées sous chaque classe et type de métrique.

## Index

- [«UC \(unités de traitement centrales de la plateforme\)»](#), à la page 306
- [«DISK \(magasins de données persistantes de plateforme\)»](#), à la page 307
- [«STATMQI \(statistiques d'utilisation de l'API\)»](#), à la page 308
- [«STATQ \(statistiques d'utilisation de l'API par file d'attente\)»](#), à la page 310
- [«STATAPP \(statistiques d'utilisation par application\)»](#), à la page 311
- [«NHAREPLICA \(statistiques Native HA par instance\)»](#), à la page 312

  Voir [«Surveillance de l'utilisation des ressources système à l'aide de la commande amqsrua»](#), à la page 302 pour plus d'informations sur la façon dont vous collectez les données pour les options répertoriées, à l'exception de NHAREPLICA.

Vous pouvez également utiliser la [commande ALTER QMGR](#) pour surveiller STATMQI et STATQ au niveau du gestionnaire de files d'attente, ou l'attribut de file d'attente locale STATQ pour surveiller des files d'attente individuelles ; voir [ALTER QUEUES](#) pour cette option.

## UC (unités de traitement centrales de la plateforme)

### Introduction

Lorsque les statistiques font référence à l' *intervalle en cours*, il s'agit de l'intervalle défini dans le message publié par le paramètre MQIAMO64\_MONITOR\_INTERVAL .

Les statistiques sont généralement publiées toutes les 10 secondes, l' *intervalle publié*, tant qu'il y a au moins un abonné actif, mais l'intervalle précis doit toujours être extrait du message.

**Important :** Sauf indication contraire, les métriques sont des valeurs absolues pour le point dans le temps lorsqu'elles sont capturées.

### SystemSummary (Performances de l'unité centrale-sur l'ensemble de la plateforme)

Pourcentage de temps UC utilisateur X%

Pourcentage moyen de temps (utilisé au cours du dernier intervalle de 10 secondes) utilisé par l'unité centrale lorsqu'elle était dans un code non privilégié.

Pourcentage de temps UC système X%

Pourcentage moyen de temps (pris au cours du dernier intervalle de 10 secondes) utilisé par l'unité centrale lorsqu'elle était en code privilégié.

Charge de l'UC-moyenne sur une minute X

Moyenne de charge d'une minute. La "moyenne de charge" est un terme à l'échelle de l'industrie, mais la valeur exacte indiquée peut varier d'une plateforme à l'autre.

Charge de l'UC-moyenne sur cinq minutes X

Moyenne de charge de cinq minutes. La "moyenne de charge" est un terme à l'échelle de l'industrie, mais la valeur exacte indiquée peut varier d'une plateforme à l'autre.

Charge de l'UC-moyenne sur quinze minutes X

La moyenne de charge de quinze minutes. La "moyenne de charge" est un terme à l'échelle de l'industrie, mais la valeur exacte indiquée peut varier d'une plateforme à l'autre.

Récapitulatif du système d'UC

Pourcentage de mémoire RAM disponible X%

Nombre total d'octets de mémoire RAM XMo

Pourcentage de mémoire RAM disponible X%

Nombre total d'octets de mémoire RAM XMo

### **QMGrSummary (performances de l'unité centrale-gestionnaire de files d'attente en cours d'exécution)**

Temps UC utilisateur-estimation en pourcentage pour le gestionnaire de files d'attente X%

Pourcentage de temps moyen (utilisé au cours du dernier intervalle de 10 secondes) utilisé par l'UC lorsque les processus de ce gestionnaire de files d'attente étaient dans un code non privilégié.

Temps UC système-estimation du pourcentage pour le gestionnaire de files d'attente X%

Pourcentage de temps moyen (utilisé au cours du dernier intervalle de 10 secondes) utilisé par l'unité centrale lorsque les processus de ce gestionnaire de files d'attente étaient en code privilégié.

Nombre total d'octets de mémoire RAM-estimation pour le gestionnaire de files d'attente XMo

Il s'agit d'une approximation de la mémoire utilisée par le gestionnaire de files d'attente.

### **DISK (magasins de données persistantes de plateforme)**

*SystemSummary* et *QMGrSummary* sont des valeurs absolues au moment de la capture. Voir [Introduction](#) pour plus de détails sur l' *intervalle publié*.

#### **SystemSummary (utilisation du disque-à l'échelle de la plateforme)**

Système de fichiers d'erreurs MQ -octets utilisés XMo

MQ erreurs système de fichiers-espace disponible X%

MQ Nombre de fichiers FDC X

Système de fichiers de trace MQ -octets utilisés XMo

MQ , système de fichiers de trace-espace disponible X%

#### **QMGrSummary (utilisation du disque-gestionnaires de files d'attente en cours d'exécution)**

Système de fichiers du gestionnaire de files d'attente-octets utilisés XMo

Espace disponible du système de fichiers du gestionnaire de files d'attente X%

#### **Journal (utilisation du disque-journal de reprise du gestionnaire de files d'attente)**

Octets de journal utilisés X

Nombre maximal d'octets de journal X

Nombre maximal d'octets pouvant être écrits dans le journal si toutes les extensions principale et secondaire étaient saturées. Cette valeur est inférieure à la taille du système de fichiers journal

Système de fichiers journaux-octets utilisés X

Système de fichiers journaux-octets max X

Octets physiques de journal écrits pour l'intervalle en cours X.

Voir [Introduction](#) pour la définition de l' *intervalle en cours*.

Octets logiques de journal écrits pour l' *intervalle en cours* X

Temps d'attente d'écriture de journal X uSec

Moyenne mobile représentant le temps nécessaire à une écriture unique sur le disque.

Où **LogWriteIntegrity**=*TripleWrite*, le nombre physique d'octets écrits sur le disque est supérieur aux octets logiques écrits.

Taille d'écriture de journal X, également moyenne mobile.

Journal occupé par les extensions en attente d'archivage X.

Publié uniquement lorsque **logtype**= linéaire et **LogManagement** = archive. Pour plus d'informations, voir [Strophe de journal du fichier qm.ini](#) .

Espace de journal en Mo requis pour la reprise sur incident X.

Publié uniquement lorsque **logtype**= linéaire.

Journal-espace en Mo occupé par les extensions réutilisables X

Publié uniquement lorsque **logtype**= linéaire et **LogManagement** = automatique. Pour plus d'informations, voir Strophe de journal du fichier qm.ini .

Journal-espace principal en cours d'utilisation X%.

Espace de fichier journal utilisé en tant que pourcentage des journaux principaux. Cette valeur peut être supérieure à 100%.

Utilisation de l'espace principal de la charge de travail du journal X%.

Pourcentage d'espace de fichier journal utilisé comme moyenne mobile par rapport à l'historique récent.

## STATMQI (statistiques d'utilisation de l'API)

Toutes les statistiques d'utilisation de l'API reflètent les occurrences et / ou les pourcentages de l'intervalle publié. Voir Introduction pour la définition de l' *intervalle publié*.

Les statistiques affichent un comptage du nombre d'appels MQI ayant échoué, mais tous les appels MQI ayant échoué n'apparaissent pas dans ces statistiques. En effet, les échecs de tous les appels MQI n'ont pas leurs statistiques enregistrées. En effet, de nombreuses raisons pour lesquelles les appels MQI échouent sont diagnostiquées avant que l'appel MQI n'atteigne les éléments internes du gestionnaire de files d'attente dans lequel les statistiques sont enregistrées.

Par exemple, MQRC\_HCONN\_ERROR est renvoyé à une application client. Si une application client transmet une erreur **hconn**, le client MQ diagnostique cette erreur et renvoie MQRC\_HCONN\_ERROR sans transmettre l'appel MQI au gestionnaire de files d'attente. Par conséquent, l'appel MQI ayant échoué n'apparaît jamais dans les statistiques enregistrées par le gestionnaire de files d'attente.

Les statistiques des appels MQI ayant échoué sont utiles car elles permettent aux clients d'identifier et de résoudre les problèmes liés aux applications incorrectement écrites qui génèrent des appels MQI ayant échoué inutilement, ce qui a un impact sur les performances. Voici quelques exemples de raisons de l'échec de divers appels MQI enregistrés dans les statistiques:

- MQCONN/MQCONN/MQOPEN renvoie 2035 MQRC\_NOT\_AUTHORIZED lorsqu'il est diagnostiqué par le gestionnaire de files d'attente et non par le client. Par exemple, en exécutant **amqspu**t en tant que personne.
- MQPUT/MQPUT1 renvoie 2053 MQRC\_Q\_FULL car MAXDEPTH a été dépassé.
- MQGET renvoie 2033 MQRC\_NO\_MSG\_AVAILABLE lors de la recherche ou de l'extraction destructive à partir d'une file d'attente vide
- MQSUBRQ renvoie 2437 MQRC\_NO\_RETAINED\_MSG car il n'y a pas de message conservé

### CONNDISC (MQCONN et MQDISC)

Nombre de MQCONN/MQCONN X

Nombre d'échecs MQCONN/MQCONN X

Connexions simultanées-cote d'alerte haute X

Nombre de MQDISC X

### OPENCLOSE (MQOPEN et MQCLOSE)

Nombre de MQOPEN X Ypar seconde

Nombre d'échecs MQOPEN X

Nombre de MQCLOSE X Y/ sec

Nombre d'échecs MQCLOSE X

### INQSET (MQINQ et MQSET)

Nombre de MQINQ X

Nombre d'échecs MQINQ X

Nombre de MQSET X

Nombre de MQSET ayant échoué X

**PUT (MQPUT)**

Nombre total d'intervalles MQPUT/MQPUT1 *X*  
Nombre total d'intervalles MQPUT/MQPUT1 nombre d'octets *X Y/ sec*  
Nombre de MQPUT de message non persistant *X*  
Nombre de MQPUT de message persistant *X*  
Nombre d'échecs MQPUT *X*  
Nombre de messages non persistants MQPUT1 *X*  
Message persistant MQPUT1 nombre *X*  
Echec de MQPUT1 nombre *X*  
Messages non persistants insérés-Nombre d'octets *X Y/ sec*  
Messages persistants insérés-Nombre d'octets *X*  
Nombre de MQSTAT *X*

**GET (MQGET)**

Nombre total d'extractions destructives de l'intervalle *X*  
Nombre total d'octets d'extraction destructifs de l'intervalle *X Y/ sec*  
Nombre d'extractions destructives de messages non persistants *X*  
Nombre d'extractions destructives de messages persistants *X*  
Echec de MQGET-nombre *X*  
Nombre d'octets de messages non persistants extraits *X Y/ sec*  
Messages persistants extraits-Nombre d'octets *X*  
Nombre de recherches de messages non persistants *X*  
Nombre de recherches de messages persistants *X*  
Nombre de recherches ayant échoué *X*  
Nombre d'octets de navigation dans les messages non persistants *X Y/ sec*  
Nombre d'octets de l'exploration des messages persistants *X*  
Nombre de messages arrivés à expiration *X*  
Nombre de files d'attente purgées *X*  
Nombre de MQCB *X*  
Nombre d'échecs MQCB *X*  
Nombre de MQCTL *X*

**SYNCPOINT (validation et annulation)**

Nombre de validations *X*  
Nombre d'annulations *X*

**SUBSCRIBE (abonnement)**

Créer un nombre d'abonnements durables *X*  
Modifier le nombre d'abonnements durables *X*  
Reprendre le nombre d'abonnements durables *X*  
Créer un nombre d'abonnements non durables *X*  
Echec de la création / alter/reprise du nombre d'abonnements *X*  
Supprimer le nombre d'abonnements durables *X*  
Supprimer le nombre d'abonnements non durables *X*  
Nombre d'échecs de suppression d'abonnement *X*  
Nombre de MQSUBRQ *X*  
Nombre d'échecs MQSUBRQ *X*

Abonné durable-cote d'alerte haute X  
Abonné durable-cote d'alerte basse X  
Abonné non durable-cote d'alerte haute X  
Abonné non durable-cote d'alerte basse X

### **PUBLISH (publier)**

Rubrique MQPUT/MQPUT1 intervalle total X  
Nombre total d'octets de rubrique insérés dans l'intervalle X Y/ sec  
Publié pour les abonnés-nombre de messages X  
Publié pour les abonnés-nombre d'octets X  
Non persistant-topic MQPUT/MQPUT1 nombre X  
Persistant-topic MQPUT/MQPUT1 nombre X  
Rubrique MQPUT/MQPUT1 nombre X

### **STATQ (statistiques d'utilisation de l'API par file d'attente)**

#### **GENERAL (Général)**

messages arrivés à expiration X (déplacés de GET for IBM MQ 9.3.0 et versions ultérieures CD )  
Nombre de files d'attente purgées X (déplacées depuis GET for IBM MQ 9.3.0 et versions ultérieures CD )  
temps moyen en file d'attente X uSec (transféré de GET for IBM MQ 9.3.0 et versions CD ultérieures)  
Nombre de lignes de la file d'attente X (déplacé de GET for IBM MQ 9.3.0 et versions ultérieures CD )

#### **OPENCLOSE (MQOPEN et MQCLOSE)**

Nombre de MQOPEN X  
Nombre de MQCLOSE X

#### **INQSET (MQINQ et MQSET)**

Nombre de MQINQ X  
Nombre de MQSET X

#### **PUT (MQPUT et MQPUT1)**

MQPUT/MQPUT1 nombre X  
Nombre d'octets MQPUT X  
Nombre de messages non persistants MQPUT X  
Nombre de messages persistants MQPUT X  
nombre MQPUT annulé X  
MQPUT1 nombre de messages non persistants X  
MQPUT1 nombre de messages persistants X  
Nombre d'octets non persistants X  
nombre d'octets persistants X  
conflits d'accès X%

Pourcentage de tentatives de verrouillage de la file d'attente ayant entraîné l'attente d'un autre processus pour libérer le verrou en premier. La diminution des conflits d'accès est susceptible d'augmenter le débit maximal de votre système car la prise d'un verrou qui n'est pas actuellement verrouillé est plus efficace que l'attente de la libération d'un verrou.

file d'attente des insertions évitées X%

Si un message est inséré dans une file d'attente alors qu'il y a une méthode d'accès get en attente, il se peut que le message n'ait pas besoin d'être mis en file d'attente car il peut être transmis immédiatement à la méthode d'accès get. Ce message est donc dit avoir évité la file d'attente, et "file d'attente évitée puts" est le nombre de ces messages. L'augmentation de l'évitement de file d'attente

est susceptible d'augmenter le débit maximal de votre système car elle évite le coût de l'insertion du message dans la file d'attente et de sa remise hors tension.

octets évités dans la file d'attente X%

Si un message est inséré dans une file d'attente alors qu'il y a une méthode d'accès get en attente, il se peut que le message n'ait pas besoin d'être mis en file d'attente car il peut être transmis immédiatement à la méthode d'accès get. Ce message est donc dit avoir évité la file d'attente, et "queue ??? bytes" est le nombre de ces octets. L'augmentation de l'évitement de file d'attente est susceptible d'augmenter le débit maximal de votre système car elle évite le coût de l'insertion du message dans la file d'attente et de sa remise hors tension.

## **GET (MQGET)**

Nombre de demandes MQGET X

Nombre d'octets MQGET X

Nombre de messages non persistants MQGET destructifs X

Nombre de messages persistants MQGET destructifs X

nombre MQGET annulé X

Nombre d'octets non persistants MQGET destructifs X

Nombre d'octets persistants MQGET destructifs X

MQGET parcourez le nombre de messages non persistants X

Nombre de messages persistants MQGET browse X

MQGET parcourez le nombre d'octets non persistants X

Nombre d'octets persistants MQGET browse X

messages arrivés à expiration X (déplacés vers GENERAL à partir de IBM MQ 9.3)

Nombre de files d'attente purgées X (déplacées vers GENERAL à partir de IBM MQ 9.3)

temps moyen en file d'attente X uSec (déplacé vers GENERAL depuis IBM MQ 9.3)

Longueur de la file d'attente X (déplacé vers GENERAL depuis IBM MQ 9.3)

Echec de la commande destructive MQGET X

Echec de la commande MQGET destructive avec MQRC\_NO\_MSG\_AVAILABLE X

Echec de la requête MQGET destructive avec MQRC\_TRUNCATED\_MSG\_FAILED X

Echec de l'exploration MQGET X

Echec de l'exploration MQGET avec MQRC\_NO\_MSG\_AVAILABLE X

Echec de l'exploration MQGET avec MQRC\_TRUNCATED\_MSG\_FAILED X

## **STATAPP (statistiques d'utilisation par application)**

### **INSTANCE (statistiques d'instance)**

Nombre d'instances X absolu

Nombre d'instances mobiles X absolu

Nombre de manques d'instance X absolu

Instances démarrées X intervalle

Déplacements de l'instance sortante initiée X intervalle

Déplacements d'instance sortants terminés X intervalle

Instances terminées lors de la reconnexion X intervalle

Instances terminées X intervalle

## NHAREPLICA (statistiques Native HA par instance)

### REPLICATION (statistiques de réplication)

Octets de journal synchrones envoyés  $X$

Octets de journal de rattrapage envoyés  $X$

▶ V 9.4.0 Octets de journal compressés synchrones envoyés  $X$

▶ V 9.4.0 Octets de journal compressé de rattrapage envoyés  $X$

▶ V 9.4.0 Octets de journal non compressés synchrones envoyés  $X$

▶ V 9.4.0 Octets de journal non compressé de rattrapage envoyés  $X$

Temps d'attente moyen d'accusé de réception d'écriture de journal  $X$   $\mu$ Sec

Taille moyenne d'accusé de réception d'écriture de journal  $X$

Octets en attente  $X$

Nombre moyen d'octets en attente  $X$

### Information associée

Métriques publiées lors de l'utilisation de IBM MQ Operator

## Surveillance en temps réel

La surveillance en temps réel est une technique qui permet de déterminer l'état en cours des files d'attente et des canaux dans un gestionnaire de files d'attente. Les informations renvoyées sont exactes au moment où la commande a été émise.

Un certain nombre de commandes sont disponibles qui, lorsqu'elles sont émises, renvoient des informations en temps réel sur les files d'attente et les canaux. Les informations peuvent être renvoyées pour une ou plusieurs files d'attente ou canaux et peuvent varier en quantité. La surveillance en temps réel peut être utilisée dans les tâches suivantes:

- Aider les administrateurs système à comprendre l'état stabilisé de leur système IBM MQ . Cela permet de diagnostiquer les problèmes si un problème se produit dans le système.
- Détermination de la condition de votre gestionnaire de files d'attente à tout moment, même si aucun événement ou problème spécifique n'a été détecté.
- Aide à déterminer la cause d'un problème dans votre système.

Avec la surveillance en temps réel, les informations peuvent être renvoyées pour les files d'attente ou les canaux. La quantité d'informations en temps réel renvoyées est contrôlée par les attributs de gestionnaire de files d'attente, de file d'attente et de canal.

- Vous pouvez surveiller une file d'attente en émettant des commandes pour vous assurer que la file d'attente est correctement traitée. Avant de pouvoir utiliser certains attributs de file d'attente, vous devez les activer pour la surveillance en temps réel.
- Vous pouvez surveiller un canal en émettant des commandes pour vous assurer que le canal s'exécute correctement. Avant de pouvoir utiliser certains attributs de canal, vous devez les activer pour la surveillance en temps réel.

La surveillance en temps réel des files d'attente et des canaux s'ajoute à la surveillance des performances et des événements de canal et est distincte de cette surveillance.

## Attributs contrôlant la surveillance en temps réel

Certains attributs de statut de file d'attente et de canal contiennent des informations de surveillance, si la surveillance en temps réel est activée. Si la surveillance en temps réel n'est pas activée, aucune information de surveillance n'est conservée dans ces attributs de surveillance. Des exemples montrent comment utiliser ces attributs de statut de file d'attente et de canal.



Vous pouvez activer ou désactiver la surveillance en temps réel pour des files d'attente ou des canaux individuels ou pour plusieurs files d'attente ou canaux. Pour contrôler des files d'attente ou des canaux individuels, définissez l'attribut de file d'attente MONQ ou l'attribut de canal MONCHL pour activer ou désactiver la surveillance en temps réel. Pour contrôler plusieurs files d'attente ou canaux ensemble, activez ou désactivez la surveillance en temps réel au niveau du gestionnaire de files d'attente à l'aide des attributs de gestionnaire de files d'attente MONQ et MONCHL. Pour tous les objets file d'attente et canal avec un attribut de surveillance spécifié avec la valeur par défaut, QMGR, la surveillance en temps réel est contrôlée au niveau du gestionnaire de files d'attente.

Les canaux émetteurs de cluster définis automatiquement ne sont pas des objets IBM MQ . Par conséquent, ils ne possèdent pas d'attributs de la même manière que les objets de canal. Pour contrôler les canaux émetteurs de cluster définis automatiquement, utilisez l'attribut de gestionnaire de files d'attente MONACLS. Cet attribut détermine si les canaux émetteurs de cluster définis automatiquement dans un gestionnaire de files d'attente sont activés ou désactivés pour la surveillance des canaux.

Pour la surveillance en temps réel des canaux, vous pouvez définir l'attribut MONCHL sur l'un des trois niveaux de surveillance: faible, moyen ou élevé. Vous pouvez définir le niveau de surveillance au niveau de l'objet ou au niveau du gestionnaire de files d'attente. Le choix du niveau dépend de votre système. La collecte de données de surveillance peut nécessiter des instructions relativement coûteuses en termes de calcul, telles que l'obtention de l'heure système. Pour réduire l'effet de la surveillance en temps réel, les options de surveillance moyenne et faible mesurent un échantillon de données à intervalles réguliers plutôt que de collecter des données en permanence. Le [Tableau 30](#), à la [page 313](#) récapitule les niveaux de surveillance disponibles pour la surveillance en temps réel des canaux:

<i>Tableau 30. Niveaux de surveillance</i>		
<b>Niveau</b>	<b>Description</b>	<b>Utilisation</b>
Bas	Mesurez un petit échantillon de données, à intervalles réguliers.	Pour les objets qui traitent un volume élevé de messages.
Moyen	Mesurez un échantillon de données, à intervalles réguliers.	Pour la plupart des objets.
Elevé	Mesurez toutes les données à intervalles réguliers.	Pour les objets qui ne traitent que quelques messages par seconde, sur lesquels les informations les plus récentes sont importantes.

Pour la surveillance en temps réel des files d'attente, vous pouvez définir l'attribut MONQ sur l'un des trois niveaux de surveillance, faible, moyen ou élevé. Cependant, il n'y a pas de distinction entre ces valeurs. Les valeurs activent toutes la collecte de données, mais n'affectent pas la taille de l'échantillon.

## Exemples

Les exemples suivants montrent comment définir les attributs de file d'attente, de canal et de gestionnaire de files d'attente nécessaires pour contrôler le niveau de surveillance. Pour tous les exemples, lorsque la surveillance est activée, les objets de file d'attente et de canal ont un niveau moyen de surveillance.

1. Pour activer la surveillance des files d'attente et des canaux pour toutes les files d'attente et tous les canaux au niveau du gestionnaire de files d'attente, utilisez les commandes suivantes:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(QMGR)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(QMGR)
```

2. Pour activer la surveillance de toutes les files d'attente et de tous les canaux, à l'exception de la file d'attente locale, Q1, et du canal émetteur, QM1 . TO . QM2, utilisez les commandes suivantes:

```
ALTER QMGR MONQ(MEDIUM) MONCHL(MEDIUM)
ALTER QL(Q1) MONQ(OFF)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(OFF)
```

3. Pour désactiver la surveillance des files d'attente et des canaux pour toutes les files d'attente et tous les canaux, à l'exception de la file d'attente locale, Q1, et du canal émetteur, QM1 . TO . QM2, utilisez les commandes suivantes:

```
ALTER QMGR MONQ(OFF) MONCHL(OFF)
ALTER QL(Q1) MONQ(MEDIUM)
ALTER CHL(QM1.TO.QM2) CHLTYPE(SDR) MONCHL(MEDIUM)
```

4. Pour désactiver la surveillance des files d'attente et des canaux pour toutes les files d'attente et tous les canaux, quels que soient les attributs d'objet individuels, utilisez la commande suivante:

```
ALTER QMGR MONQ(NONE) MONCHL(NONE)
```

5. Pour contrôler les fonctions de surveillance des canaux émetteurs de cluster définis automatiquement, utilisez la commande suivante:

```
ALTER QMGR MONACLS(MEDIUM)
```

6. Pour indiquer que les canaux émetteurs de cluster définis automatiquement doivent utiliser le paramètre de gestionnaire de files d'attente pour la surveillance des canaux, utilisez la commande suivante:

```
ALTER QMGR MONACLS(QMGR)
```

### Concepts associés

«Surveillance en temps réel», à la page 312

La surveillance en temps réel est une technique qui permet de déterminer l'état en cours des files d'attente et des canaux dans un gestionnaire de files d'attente. Les informations renvoyées sont exactes au moment où la commande a été émise.

«Using IBM MQ online monitoring», à la page 334

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

### Tâches associées

«Affichage des données de surveillance des files d'attente et des canaux», à la page 314

Pour afficher des informations de surveillance en temps réel pour une file d'attente ou un canal, utilisez la commande IBM MQ Explorer ou la commande MQSC appropriée. Certaines zones de surveillance affichent une paire de valeurs d'indicateur séparées par des virgules, qui vous aident à surveiller le fonctionnement de votre gestionnaire de files d'attente. Des exemples montrent comment afficher les données de surveillance.

Affichage et modification des attributs du gestionnaire de files d'attente

Surveillance (MONCHL)

## Affichage des données de surveillance des files d'attente et des canaux

Pour afficher des informations de surveillance en temps réel pour une file d'attente ou un canal, utilisez la commande IBM MQ Explorer ou la commande MQSC appropriée. Certaines zones de surveillance affichent une paire de valeurs d'indicateur séparées par des virgules, qui vous aident à surveiller le fonctionnement de votre gestionnaire de files d'attente. Des exemples montrent comment afficher les données de surveillance.

## Pourquoi et quand exécuter cette tâche

Les zones de surveillance qui affichent une paire de valeurs séparées par une virgule fournissent des indicateurs à court terme et à long terme pour la durée mesurée depuis l'activation de la surveillance pour l'objet ou depuis le démarrage du gestionnaire de files d'attente:

- L'indicateur à court terme est la première valeur du couple et est calculé de telle sorte que les mesures plus récentes reçoivent une pondération plus élevée et auront un effet plus important sur cette valeur. Ceci donne une indication de l'évolution récente des mesures effectuées.
- L'indicateur à long terme dans la deuxième valeur du couple et est calculé de telle sorte que les mesures plus récentes ne reçoivent pas une pondération aussi élevée. Cela donne une indication de l'activité à plus long terme sur les performances d'une ressource.

Ces valeurs d'indicateur sont particulièrement utiles pour détecter les changements dans le fonctionnement de votre gestionnaire de files d'attente. Cela nécessite une connaissance des temps que ces indicateurs montrent en utilisation normale, afin de détecter des augmentations dans ces temps. En collectant et en vérifiant régulièrement ces valeurs, vous pouvez détecter des fluctuations dans le fonctionnement de votre gestionnaire de files d'attente. Cela peut indiquer une modification des performances.

Obtenez des informations de surveillance en temps réel comme suit:

## Procédure

1. Pour afficher des informations de surveillance en temps réel pour une file d'attente, utilisez la IBM MQ Explorer ou la commande `MQSC DISPLAY QSTATUS`, en spécifiant le paramètre facultatif `MONITOR`.
2. Pour afficher des informations de surveillance en temps réel pour un canal, utilisez la IBM MQ Explorer ou la commande `MQSC DISPLAY CHSTATUS`, en spécifiant le paramètre facultatif `MONITOR`.

## Exemple

La file d'attente, `Q1`, a l'attribut `MONQ` défini sur la valeur par défaut, `QMGR`, et le gestionnaire de files d'attente propriétaire de la file d'attente a l'attribut `MONQ` défini sur `MEDIUM`. Pour afficher les zones de surveillance collectées pour cette file d'attente, utilisez la commande suivante:

```
DISPLAY QSTATUS(Q1) MONITOR
```

Les zones de surveillance et le niveau de surveillance de la file d'attente, `Q1`, s'affichent comme suit:

```
QSTATUS(Q1)
TYPE(Queue)
MONQ(MEDIUM)
QTIME(11892157,24052785)
MSGAGE(37)
LPUTDATE(2005-03-02)
LPUTTIME(09.52.13)
LGETDATE(2005-03-02)
LGETTIME(09.51.02)
```

Le canal émetteur, `QM1.TO.QM2`, a l'attribut `MONCHL` défini sur la valeur par défaut, `QMGR`, et le gestionnaire de files d'attente propriétaire de la file d'attente a l'attribut `MONCHL` défini sur `MEDIUM`. Pour afficher les zones de surveillance collectées pour ce canal émetteur, utilisez la commande suivante:

```
DISPLAY CHSTATUS(QM1.TO.QM2) MONITOR
```

Les zones de surveillance et le niveau de surveillance du canal émetteur, `QM1.TO.QM2`, s'affichent comme suit:

```
CHSTATUS(QM1.TO.QM2)
XMITQ(Q1)
CONNAM(127.0.0.1)
CURRENT
```

```
CHLTYPE (SDR)
STATUS (RUNNING)
SUBSTATE (MQGET)
MONCHL (MEDIUM)
XQTIME (755394737, 755199260)
NETTIME (13372, 13372)
EXITTIME (0, 0)
XBATCHSZ (50, 50)
COMPTIME (0, 0)
STOPREQ (NO)
RQMNAME (QM2)
```

### Concepts associés

«Surveillance en temps réel», à la page 312

La surveillance en temps réel est une technique qui permet de déterminer l'état en cours des files d'attente et des canaux dans un gestionnaire de files d'attente. Les informations renvoyées sont exactes au moment où la commande a été émise.

### Référence associée

STATUT D'AFFICHAGE

## Surveillance des files d'attente

Utilisez cette page pour afficher les tâches qui vous aident à résoudre un problème lié à une file d'attente et à l'application qui la dessert. Différentes options de surveillance sont disponibles pour déterminer le problème

Souvent, le premier signe d'un problème avec une file d'attente en cours de traitement est que le nombre de messages dans la file d'attente (CURDEPTH) augmente. Si vous prévoyez une augmentation à certains moments de la journée ou sous certaines charges de travail, un nombre croissant de messages peut ne pas indiquer un problème. Toutefois, si vous n'avez pas d'explication à l'augmentation du nombre de messages, vous pouvez en rechercher la cause.

Vous pouvez avoir une file d'attente d'application dans laquelle il y a un problème avec l'application ou une file d'attente de transmission dans laquelle il y a un problème avec le canal. Des options de surveillance supplémentaires sont disponibles lorsque l'application qui dessert la file d'attente est un canal.

Les exemples suivants examinent les problèmes liés à une file d'attente particulière, appelée Q1, et décrivent les zones que vous examinez dans la sortie de diverses commandes:

### ***Détermination de l'ouverture de la file d'attente de votre application***

Si vous rencontrez un problème avec une file d'attente, vérifiez si la file d'attente est ouverte dans votre application

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

Pour déterminer si la file d'attente de votre application est ouverte, procédez comme suit:

#### **Procédure**

1. Assurez-vous que l'application qui s'exécute sur la file d'attente est celle que vous attendez. Exécutez la commande suivante pour la file d'attente en question:

```
DISPLAY QSTATUS(Q1) TYPE(HANDLE) ALL
```

Dans la sortie, examinez la zone APPLTAG et vérifiez que le nom de votre application est affiché. Si le nom de votre application n'est pas affiché ou s'il n'y a pas de sortie du tout, démarrez votre application.

2. Si la file d'attente est une file d'attente de transmission, consultez la sortie dans la zone CHANNEL. Si le nom de canal n'apparaît pas dans la zone CHANNEL, déterminez si le canal est en cours d'exécution.

3. Vérifiez que la file d'attente de l'application qui s'exécute sur la file d'attente est ouverte en entrée. Entrez la commande suivante :

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Dans la sortie, examinez la zone IPPROCS pour voir si une application dispose d'une file d'attente ouverte pour l'entrée. Si la valeur est 0 et qu'il s'agit d'une file d'attente d'application utilisateur, assurez-vous que l'application ouvre la file d'attente en entrée pour extraire les messages de la file d'attente.

### ***Vérification de la disponibilité des messages dans la file d'attente***

Si vous avez un grand nombre de messages dans la file d'attente et que votre application ne traite aucun de ces messages, vérifiez si les messages de la file d'attente sont disponibles pour votre application

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

Pour déterminer la raison pour laquelle votre application ne traite pas les messages de la file d'attente, procédez comme suit:

### **Procédure**

1. Vérifiez que votre application ne demande pas d'ID de message ou d'ID de corrélation spécifique lorsqu'elle doit traiter tous les messages de la file d'attente.
2. Bien que la longueur en cours de la file d'attente puisse indiquer qu'il y a un nombre croissant de messages dans la file d'attente, il se peut que certains messages de la file d'attente ne soient pas disponibles pour être obtenus par une application, car ils ne sont pas validés ; la longueur en cours inclut le nombre de MQPUT de messages non validés dans la file d'attente. Entrez la commande suivante :

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Dans la sortie, examinez la zone UNCOM pour voir s'il existe des messages non validés dans la file d'attente.

3. Si votre application tente d'extraire des messages de la file d'attente, vérifiez si l'application d'insertion valide correctement les messages. Exécutez la commande suivante pour trouver les noms des applications qui placent des messages dans cette file d'attente:

```
DISPLAY QSTATUS(Q1) TYPE(Handle) OPENTYPE(Output)
```

4. Exécutez ensuite la commande suivante, en insérant dans *appltag* la valeur APPLTAG à partir de la sortie de la commande précédente:

```
DISPLAY CONN(*) WHERE(APPLTAG EQ appltag) UOWSTDA UOWSTTI
```

Indique quand l'unité de travail a été démarrée et vous aide à déterminer si l'application crée une unité de travail de longue durée. Si l'application d'insertion est un canal, vous souhaitez peut-être déterminer pourquoi l'exécution d'un lot prend beaucoup de temps.

### ***Vérification de l'extraction des messages de la file d'attente par votre application***

Si vous rencontrez un problème avec une file d'attente et l'application qui la dessert, vérifiez si votre application obtient des messages de la file d'attente

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

Pour vérifier si votre application extrait des messages de la file d'attente, procédez comme suit:

## Procédure

1. Vérifiez que l'application qui s'exécute sur la file d'attente traite effectivement les messages de la file d'attente. Entrez la commande suivante :

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Dans la sortie, examinez les zones LGETDATE et LGETTIME qui indiquent quand la dernière extraction a été effectuée à partir de la file d'attente.

2. Si la dernière extraction de cette file d'attente remonte à plus longtemps que prévu, vérifiez que l'application traite correctement les messages.

Si l'application est un canal, vérifiez si les messages passent par ce canal

### ***Déterminer si l'application peut traiter les messages assez rapidement***

Si des messages sont générés dans la file d'attente, mais que vos autres vérifications n'ont pas détecté de problèmes de traitement, vérifiez que l'application peut traiter les messages assez rapidement. Si l'application est un canal, vérifiez que le canal peut traiter les messages assez rapidement.

## Pourquoi et quand exécuter cette tâche

Pour déterminer si l'application traite les messages assez rapidement, effectuez les tests suivants:

## Procédure

1. Exécutez la commande suivante régulièrement pour collecter des données de performances sur la file d'attente:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) ALL
```

Si les valeurs des indicateurs QTIME sont élevées ou augmentent au cours de la période, et que vous avez déjà exclu la possibilité d'unités de travail à exécution longue en vérifiant que les messages de la file d'attente sont disponibles, l'application d'extraction risque de ne pas suivre les applications d'insertion.

2. Si votre application d'extraction ne peut pas suivre les applications d'insertion, envisagez d'ajouter une autre application d'extraction pour traiter la file d'attente.

La possibilité d'ajouter une autre application d'extraction dépend de la conception de l'application et de la possibilité de partager la file d'attente avec plusieurs applications. Des fonctions telles que le regroupement de messages ou l'obtention par ID de corrélation peuvent aider à garantir que deux applications peuvent traiter une file d'attente simultanément.

### ***Vérification de la file d'attente lorsque la longueur en cours n'augmente pas***

Même si la longueur en cours de votre file d'attente n'augmente pas, il peut être utile de surveiller la file d'attente pour vérifier si votre application traite correctement les messages.

## Pourquoi et quand exécuter cette tâche

Pour collecter des données de performances sur la file d'attente, émettez régulièrement la commande suivante:

## Procédure

Exécutez régulièrement la commande suivante:

```
DISPLAY QSTATUS(Q1) TYPE(Queue) MSGAGE QTIME
```

Dans la sortie, si la valeur de MSGAGE augmente au cours de la période et que votre application est conçue pour traiter tous les messages, cela peut indiquer que certains messages ne sont pas traités du tout.

## Surveillance des canaux

Cette page permet d'afficher les tâches qui vous aident à résoudre un problème lié à une file d'attente de transmission et au canal qui la dessert. Différentes options de surveillance de canal sont disponibles pour déterminer le problème.

Souvent, le premier signe d'un problème avec une file d'attente en cours de traitement est que le nombre de messages dans la file d'attente (CURDEPTH) augmente. Si vous prévoyez une augmentation à certains moments de la journée ou sous certaines charges de travail, un nombre croissant de messages peut ne pas indiquer un problème. Toutefois, si vous n'avez pas d'explication à l'augmentation du nombre de messages, vous pouvez en rechercher la cause.

Il se peut que vous ayez un problème avec le canal qui dessert une file d'attente de transmission. Diverses options de surveillance de canal sont disponibles pour vous aider à déterminer le problème.

Les exemples suivants examinent les problèmes liés à une file d'attente de transmission appelée QM2 et à un canal appelé QM1.TO.QM2. Ce canal est utilisé pour envoyer des messages du gestionnaire de files d'attente, QM1, au gestionnaire de files d'attente, QM2. La définition de canal sur le gestionnaire de files d'attente QM1 est un canal émetteur ou serveur, et la définition de canal sur le gestionnaire de files d'attente, QM2, est un canal récepteur ou demandeur.

### **Déterminer si le canal est en cours d'exécution**

Si vous rencontrez un problème avec une file d'attente de transmission, vérifiez si le canal est en cours d'exécution.

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

Pour vérifier l'état du canal qui gère la file d'attente de transmission, procédez comme suit:

#### **Procédure**

1. Exécutez la commande suivante pour savoir quel canal vous prévoyez de traiter la file d'attente de transmission QM2:

```
DIS CHANNEL(*) WHERE(XMITQ EQ QM2)
```

Dans cet exemple, la sortie de cette commande indique que le canal gérant la file d'attente de transmission est QM1.TO.QM2 QM2

2. Exécutez la commande suivante pour déterminer le statut du canal, QM1.TO.QM2:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

3. Examinez la zone STATUS de la sortie de la commande **CHSTATUS** :
  - Si la valeur de la zone STATUS est RUNNING, vérifiez que le canal déplace les messages
  - Si la sortie de la commande n'affiche aucun statut ou si la valeur de la zone STATUS est STOPPED, RETRY, BINDING ou REQUESTING, effectuez l'étape appropriée, comme suit:
4. Facultatif : Si la valeur de la zone STATUS n'indique aucun statut, le canal est inactif. Par conséquent, procédez comme suit:
  - a) Si le canal aurait dû être démarré automatiquement par un déclencheur, vérifiez que les messages de la file d'attente de transmission sont disponibles.  
Si des messages sont disponibles dans la file d'attente de transmission, vérifiez que les paramètres de déclenchement de la file d'attente de transmission sont corrects.
  - b) Exécutez la commande suivante pour redémarrer le canal manuellement:

```
START CHANNEL(QM1.TO.QM2)
```


5. Facultatif : Si la valeur de la zone STATUS est STOPPED, procédez comme suit:

a) Consultez les journaux des erreurs pour déterminer la raison pour laquelle le canal s'est arrêté. Si le canal s'est arrêté en raison d'une erreur, corrigez le problème.

Vérifiez également que le canal a des valeurs spécifiées pour les attributs de relance: *SHORTRTY* et *LONGRTY*. En cas de pannes transitoires telles que des erreurs réseau, le canal tente alors de redémarrer automatiquement.

b) Exécutez la commande suivante pour redémarrer le canal manuellement:

```
START CHANNEL(QM1.TO.QM2)
```

 Sous IBM MQ for z/OS, vous pouvez détecter lorsqu'un utilisateur arrête un canal à l'aide de messages d'événement de commande.

6. Facultatif : Si la valeur de la zone STATUS est RETRY, procédez comme suit:

a) Consultez les journaux des erreurs pour identifier l'erreur, puis corrigez le problème.

b) Exécutez la commande suivante pour redémarrer le canal manuellement:

```
START CHANNEL(QM1.TO.QM2)
```

ou attendez que le canal se connecte correctement lors de sa prochaine tentative.

7. Facultatif : Si la valeur de la zone STATUS est BINDING ou REQUESTING, cela signifie que le canal n'a pas encore été connecté au partenaire. Procédez comme suit :

a) Exécutez la commande suivante, aux deux extrémités du canal, pour déterminer le sous-état du canal:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

**Remarque :**

i) Dans certains cas, il peut y avoir un sous-état à une extrémité du canal uniquement.


ii) De nombreux sous-états étant transitoires, émettez la commande quelques fois pour détecter si un canal est bloqué dans un sous-état particulier.

b) Vérifiez [Tableau 31](#), à la page 320 pour déterminer l'action à effectuer:

Tableau 31. Sous-états vus avec une liaison de statut ou une demande		
Lancement du sous-état MCA <sup>1</sup>	Sous-état MCA répondante <sup>2</sup>	Remarques
NameServer		L'agent MCA initiateur attend la fin d'une demande de serveur de noms. Vérifiez que le nom d'hôte correct a été spécifié dans l'attribut de canal, CONNAME, et que vos serveurs de noms sont configurés correctement.
SCYEXIT	SCYEXIT	Les MCM sont actuellement <i>en conversation</i> via un exit de sécurité. Pour plus d'informations, voir «Déterminer si le canal peut traiter les messages assez rapidement», à la page 323.
	CHADEXIT	L'exit d'autodéfinition de canal est en cours d'exécution. Pour plus d'informations, voir «Déterminer si le canal peut traiter les messages assez rapidement», à la page 323.



Tableau 31. Sous-états vus avec une liaison de statut ou une demande (suite)

Lancement du sous-état MCA <sup>1</sup>	Sous-état MCA répondante <sup>2</sup>	Remarques
RCVEXIT SENDEXIT MSGEXIT MREXIT	RCVEXIT SENDEXIT MSGEXIT MREXIT	Les exits sont appelés au démarrage du canal pour MQXR_INIT. Vérifiez le traitement dans cette partie de votre exit si cela prend beaucoup de temps. Pour plus d'informations, voir «Déterminer si le canal peut traiter les messages assez rapidement», à la page 323.
Sérialiser	Sérialiser	Ce sous-état s'applique uniquement aux canaux dont la disposition est SHARED.
NETCONNECT		Ce sous-état s'affiche si la connexion est retardée en raison d'une configuration réseau incorrecte.
SSLHANDSHAKE	SSLHANDSHAKE	Un établissement de liaison TLS se compose d'un certain nombre d'envois et de réceptions. Si les temps réseau sont lents ou si la connexion aux listes de révocation de certificat de recherche est lente, cela affecte le temps nécessaire à l'établissement de la liaison.   Sous IBM MQ for z/OS, ce sous-état peut également indiquer que le nombre de SSLTASKS est insuffisant.

#### Remarques :

- i) L'agent MCA initiateur est l'extrémité du canal qui a démarré la conversation. Il peut s'agir d'émetteurs, d'émetteurs de cluster, de serveurs qualifiés complets et de demandeurs. Dans une paire serveur-demandeur, il s'agit de la fin à partir de laquelle vous avez démarré le canal.
- ii) L'agent MCA qui répond est l'extrémité du canal qui a répondu à la demande de démarrage de la conversation. Il peut s'agir de récepteurs, de récepteurs de cluster, de demandeurs (lorsque le serveur ou l'émetteur est démarré), de serveurs (lorsque le demandeur est démarré) et d'émetteurs (dans une paire de canaux de rappel demandeur-émetteur).

#### Vérification que le canal déplace des messages

Si vous rencontrez un problème avec une file d'attente de transmission, vérifiez que le canal déplace des messages

#### Avant de commencer

Exécutez la commande `DIS CHSTATUS(QM1.TO.QM2) ALL`. Si la valeur de la zone STATUS est RUNNING, le canal s'est correctement connecté au système partenaire.

Vérifiez qu'il n'y a pas de messages non validés dans la file d'attente de transmission, comme décrit dans «Vérification de la disponibilité des messages dans la file d'attente», à la page 317.

#### Pourquoi et quand exécuter cette tâche

Si des messages sont disponibles pour le canal à extraire et à envoyer, effectuez les vérifications suivantes:

#### Procédure

1. Dans la sortie de la commande d'affichage du statut du canal, `DIS CHSTATUS(QM1.TO.QM2) ALL`, examinez les zones suivantes:

**MSGGS**

Nombre de messages envoyés ou reçus (ou, pour les canaux de connexion serveur, nombre d'appels MQI traités) au cours de cette session (depuis le démarrage du canal).

**BUFSSSENT**

Nombre de mémoires tampon de transmission envoyées. Cela inclut les transmissions pour l'envoi d'informations de contrôle uniquement.

**BYTSSSENT**

Nombre d'octets envoyés au cours de cette session (depuis le démarrage du canal). Ce paramètre inclut les informations de contrôle envoyées par l'agent MCA.

**LSTMSGDA**

Date à laquelle le dernier message a été envoyé ou l'appel MQI a été traité. Voir LSTMSGTI.

**LSTMSGTI**

Heure à laquelle le dernier message a été envoyé ou à laquelle l'appel MQI a été traité. Pour un émetteur ou un serveur, il s'agit de l'heure à laquelle le dernier message (la dernière partie du message, s'il a été divisé) a été envoyé. Pour un demandeur ou un récepteur, il s'agit de l'heure à laquelle le dernier message a été inséré dans la file d'attente cible correspondante. Pour un canal de connexion serveur, il s'agit de l'heure à laquelle le dernier appel MQI a été effectué.

**CURMSGGS**

Pour un canal émetteur, il s'agit du nombre de messages qui ont été envoyés dans le lot en cours. Pour un canal récepteur, il s'agit du nombre de messages qui ont été reçus dans le lot en cours. La valeur est remise à zéro, pour les canaux émetteurs et récepteurs, lorsque le lot est validé.

**STATUT**

Statut du canal, qui peut être Starting, Binding, Initializing, Running, Stopping, Retrying, Paused, Stoppedou Requesting.

**sous-état**

Action en cours d'exécution par le canal.

**INDOUBT**

Indique si le canal est en attente de validation. Il s'agit uniquement de YES alors que l'agent de transmission de messages d'envoi attend un accusé de réception indiquant qu'un lot de messages qu'il a envoyés a été correctement reçu. NO est indiqué dans tous les autres cas, y compris au cours de la période pendant laquelle des messages sont envoyés, mais avant qu'un accusé de réception n'ait été demandé. Pour un canal récepteur, la valeur est toujours NO.

- Déterminez si le canal a envoyé des messages depuis son démarrage. Si tel est le cas, déterminez quand le dernier message a été envoyé.
- Le canal a peut-être démarré un lot qui n'est pas encore terminé, comme indiqué par une valeur différente de zéro dans CURMSGGS. Si INDOUBT est défini sur YES, le canal attend de recevoir un accusé de réception indiquant que l'autre extrémité du canal a reçu le lot. Examinez la zone SUBSTATE dans la sortie et reportez-vous à [Tableau 32](#), à la page 322:

<i>Tableau 32. Sous-états MCA émetteur et récepteur</i>		
<b>SOUS-ETAT de l'expéditeur</b>	<b>SOUS-ETAT du récepteur</b>	<b>Remarques</b>
MQGET	RECEVOIR	Etats normaux d'un canal au repos.
ENVOYER	RECEVOIR	SEND est généralement un état transitoire. Si SEND est visible, cela indique que les mémoires tampon du protocole de communication sont remplies. Cela peut indiquer un problème de réseau.
RECEVOIR		Si l'expéditeur est vu dans le sous-état RECEIVE pendant un certain temps, il attend une réponse, soit à l'achèvement d'un lot, soit à un signal de présence. Vous pouvez être amené à vérifier pourquoi l'exécution d'un lot prend beaucoup de temps.

**Remarque :** Vous pouvez également déterminer si le canal peut traiter les messages assez rapidement, en particulier si le canal possède un sous-état associé au traitement de l'exit.

### ***Vérification de la raison pour laquelle l'exécution d'un lot prend beaucoup de temps***

Les raisons pour lesquelles un traitement par lots peut prendre beaucoup de temps incluent un réseau lent ou un canal qui utilise un traitement de relance de message.

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

Lorsqu'un canal émetteur a envoyé un lot de messages, il attend la confirmation de ce lot de la part du récepteur, sauf si le canal est pipelined. Les facteurs décrits dans cette tâche peuvent affecter la durée d'attente du canal émetteur.

### **Procédure**

- Vérifiez si le réseau est lent.

La valeur NETTIME correspond à la durée, exprimée en microsecondes, nécessaire à l'envoi d'une demande de fin de traitement par lots à l'extrémité éloignée du canal et à la réception d'une réponse moins le temps de traitement de la demande de fin de traitement par lots. Cette valeur peut être élevée pour l'une des raisons suivantes:

- Le réseau est lent. Un réseau lent peut affecter le temps nécessaire à l'exécution d'un lot. Les mesures qui aboutissent aux indicateurs de la zone NETTIME sont mesurées à la fin d'un lot. Cependant, le premier lot affecté par un ralentissement du réseau n'est pas indiqué avec une modification de la valeur NETTIME car il est mesuré à la fin du lot.
- Les demandes sont mises en file d'attente à l'extrémité distante, par exemple, un canal peut relancer une opération d'insertion, ou une demande d'insertion peut être lente en raison des E-S de l'ensemble de pages. Une fois les demandes mises en file d'attente terminées, la durée de la demande de fin de lot est mesurée. Par conséquent, si vous obtenez une valeur NETTIME élevée, recherchez un traitement inhabituel à l'extrémité distante.

- Vérifiez si le canal utilise une nouvelle tentative de message.

Si le canal récepteur ne parvient pas à placer un message dans une file d'attente cible, il peut utiliser le traitement de relance de message, au lieu de placer le message dans une file d'attente de rebut immédiatement. Une nouvelle tentative de traitement peut entraîner un ralentissement du lot. Entre les tentatives MQPUT, le canal aura STATUS (PAUSED), indiquant qu'il attend que l'intervalle entre les nouvelles tentatives de message soit écoulé.

### ***Déterminer si le canal peut traiter les messages assez rapidement***

Si des messages sont générés dans la file d'attente de transmission, mais que vous n'avez détecté aucun problème de traitement, déterminez si le canal peut traiter les messages assez rapidement.

### **Avant de commencer**

Exécutez la commande suivante à plusieurs reprises sur une période donnée pour collecter des données de performances sur le canal:

```
DIS CHSTATUS(QM1.TO.QM2) ALL
```

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

Confirmez qu'il n'y a pas de messages non validés dans la file d'attente de transmission, comme décrit dans [«Vérification de la disponibilité des messages dans la file d'attente»](#), à la page 317, puis vérifiez la zone XQTIME dans la sortie de la commande d'affichage de l'état du canal. Lorsque les valeurs des indicateurs XQTIME sont constamment élevées, ou augmentent sur la période de mesure, l'indication est que le canal ne suit pas le rythme des applications d'insertion.

Effectuez les tests suivants:

## Procédure

### 1. Vérifiez si les exits sont en cours de traitement.

Si des exits sont utilisés sur le canal qui distribue ces messages, ils peuvent être ajoutés au temps passé à traiter les messages. Pour déterminer si tel est le cas, effectuez les vérifications suivantes:

#### a) Dans la sortie de la commande DIS CHSTATUS(QM1 . TO . QM2) ALL, vérifiez la zone EXITTIME.

Si le temps passé dans les exits est plus long que prévu, recherchez dans le traitement de vos exits des boucles inutiles ou un traitement supplémentaire, en particulier dans les exits de message, d'envoi et de réception. Ce traitement affecte tous les messages déplacés sur le canal.

#### b) Dans la sortie de la commande DIS CHSTATUS(QM1 . TO . QM2) ALL, vérifiez la zone SUBSTATE.

Si le canal possède l'un des sous-états suivants depuis un certain temps, passez en revue le traitement dans vos exits:

- SCYEXIT
- RCVEXIT
- SENDEXIT
- MSGEXIT
- MREXIT

Pour plus d'informations sur les sous-états de canal, voir le tableau [Tableau 32](#), à la page 322.

### 2. Vérifiez si le réseau est lent.

Si les messages ne se déplacent pas assez rapidement sur un canal, il se peut que le réseau soit lent. Pour déterminer si tel est le cas, effectuez les vérifications suivantes:

#### a) Dans la sortie de la commande DIS CHSTATUS(QM1 . TO . QM2) ALL, vérifiez la zone NETTIME.

Ces indicateurs sont mesurés lorsque le canal émetteur demande une réponse à son partenaire. Cela se produit à la fin de chaque lot et, lorsqu'un canal est inactif lors du signal de présence.

#### b) Si cet indicateur indique que les allers-retours prennent plus de temps que prévu, utilisez d'autres outils de surveillance du réseau pour examiner les performances de votre réseau.

### 3. Vérifiez si le canal utilise la compression.

Si le canal utilise la compression, cela augmente le temps passé à traiter les messages. Si le canal utilise un seul algorithme de compression, effectuez les vérifications suivantes:

#### a) Dans la sortie de la commande DIS CHSTATUS(QM1 . TO . QM2) ALL, vérifiez la zone COMPTIME.

Ces indicateurs indiquent le temps passé pendant la compression ou la décompression.

#### b) Si la compression choisie ne réduit pas la quantité de données à envoyer par la quantité attendue, modifiez l'algorithme de compression.

### 4. Si le canal utilise plusieurs algorithmes de compression, effectuez les vérifications suivantes:

#### a) Dans la sortie de la commande DIS CHSTATUS(QM1 . TO . QM2) ALL, vérifiez les zones COMPTIME, COMPHDR et COMPMSG.

#### b) Modifiez les algorithmes de compression indiqués dans la définition de canal ou envisagez d'écrire un exit de message pour remplacer le choix de l'algorithme de compression du canal pour des messages particuliers si le taux de compression ou le choix de l'algorithme ne fournit pas la compression ou les performances requises.

## ***Résolution des problèmes liés aux canaux de cluster***

Si vous disposez d'une génération de messages sur le système SYSTEM.CLUSTER.TRANSMIT.QUEUE , la première étape du diagnostic du problème consiste à identifier le ou les canaux qui ont un problème lors de la distribution des messages.

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

Pour découvrir quel (s) canal (s), à l'aide de SYSTEM.CLUSTER.TRANSMIT.QUEUE rencontre un problème lors de la distribution des messages. Effectuez les contrôles suivants :

## Procédure

1. Entrez la commande suivante :

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

**Remarque :** Si vous disposez d'un cluster occupé qui comporte de nombreux déplacements de messages, envisagez d'exécuter cette commande avec un nombre plus élevé pour éliminer les canaux qui ne disposent que de quelques messages à distribuer.

2. Recherchez dans la sortie le ou les canaux dont les valeurs sont élevées dans la zone XQMSGSA. Déterminez la raison pour laquelle le canal ne déplace pas les messages ou ne les déplace pas assez rapidement. Utilisez les tâches décrites dans «[Surveillance des canaux](#)», à la page 319 pour diagnostiquer les problèmes liés aux canaux qui sont à l'origine de la génération.

## Surveillance des clusters

Dans un cluster, vous pouvez surveiller les messages d'application, les messages de contrôle et les journaux. Il existe des considérations de surveillance spéciales lorsque le cluster équilibre la charge entre deux ou plusieurs instances d'une file d'attente.

### Surveillance des messages d'application dans le cluster

En règle générale, tous les messages de cluster qui quittent le gestionnaire de files d'attente passent par SYSTEM.CLUSTER.TRANSMIT.QUEUE, quel que soit le canal émetteur de cluster utilisé pour transmettre le message. Chaque canal purge les messages ciblés pour ce canal en parallèle avec tous les autres canaux émetteurs de cluster. Une accumulation croissante de messages dans cette file d'attente peut indiquer un problème avec un ou plusieurs canaux et doit être examinée:

- La longueur de la file d'attente doit être surveillée de manière appropriée pour la conception du cluster.
- La commande suivante renvoie tous les canaux dont plusieurs messages sont en attente dans la file d'attente de transmission:

```
DIS CHSTATUS(*) WHERE(XQMSGSA GT 1)
```

Avec tous les messages de cluster sur une seule file d'attente, il n'est pas toujours facile de voir quel canal présente des problèmes lorsqu'il commence à se remplir. L'utilisation de cette commande est un moyen simple de voir quel canal est responsable.

Vous pouvez configurer un gestionnaire de files d'attente de cluster pour qu'il comporte plusieurs files d'attente de transmission. Si vous remplacez l'attribut de gestionnaire de files d'attente DEFCLXQ par CHANNEL, chaque canal émetteur de cluster est associé à une file d'attente de transmission de cluster différente. Vous pouvez également configurer manuellement des files d'attente de transmission distinctes. Pour afficher toutes les files d'attente de transmission de cluster associées aux canaux émetteurs de cluster, exécutez la commande suivante:

```
DISPLAY CLUSQMGR (qmgrName) XMITQ
```

Définissez les files d'attente de transmission de cluster de sorte qu'elles suivent le modèle de la racine fixe du nom de la file d'attente sur la gauche. Vous pouvez ensuite interroger la profondeur de toutes les files d'attente de transmission de cluster renvoyées par la commande **DISPLAY CLUSMGR**, à l'aide d'un nom de file d'attente générique:

```
DISPLAY QUEUE (qname *) CURDEPTH
```

## Surveillance des messages de contrôle dans le cluster

La file d'attente SYSTEM . CLUSTER . COMMAND . QUEUE est utilisée pour le traitement de tous les messages de contrôle de cluster d'un gestionnaire de files d'attente, générés par le gestionnaire de files d'attente local ou envoyés à ce gestionnaire de files d'attente à partir d'autres gestionnaires de files d'attente du cluster. Lorsqu'un gestionnaire de files d'attente gère correctement son état de cluster, cette file d'attente tend vers zéro. Dans certains cas, la longueur des messages de cette file d'attente peut augmenter temporairement:

- La présence de nombreux messages dans la file d'attente indique une attrition dans l'état du cluster.
- Lorsque vous apportez des modifications importantes, autorisez la file d'attente à se stabiliser entre ces modifications. Par exemple, lors du déplacement de référentiels, autorisez la file d'attente à atteindre zéro avant de déplacer le deuxième référentiel.

Tant qu'il existe un journal des messages en attente dans cette file d'attente, les mises à jour de l'état du cluster ou les commandes liées au cluster ne sont pas traitées. Si les messages ne sont pas supprimés de cette file d'attente depuis longtemps, des investigations supplémentaires sont nécessaires, initialement via l'inspection des journaux d'erreurs du gestionnaire de files d'attente (ou des CHINIT journaux sur z/OS ) qui peuvent expliquer le processus à l'origine de cette situation.

Le SYSTEM . CLUSTER . REPOSITORY . QUEUE contient les informations de cache du référentiel de cluster sous la forme d'un certain nombre de messages. Il est courant que les messages soient toujours présents dans cette file d'attente, et plus encore pour les clusters plus volumineux. Par conséquent, la longueur des messages de cette file d'attente n'est pas un problème.

## Surveillance des journaux

Les problèmes qui se produisent dans le cluster peuvent ne pas présenter de symptômes externes aux applications pendant de nombreux jours (et même des mois) après que le problème se soit produit initialement en raison de la mise en cache des informations et de la nature distribuée du cluster. Cependant, le problème d'origine est souvent signalé dans les IBM MQ journaux d'erreurs (et CHINIT dans z/OS). Pour cette raison, il est essentiel de surveiller activement ces journaux pour tous les messages écrits relatifs à la mise en cluster. Ces messages doivent être lus et compris, avec toutes les mesures nécessaires.

Par exemple, une interruption des communications avec un gestionnaire de files d'attente dans un cluster peut entraîner la connaissance de certaines ressources de cluster qui sont supprimées en raison de la manière dont les clusters revalident régulièrement les ressources de cluster en republiant les informations. Un avertissement indiquant qu'un tel événement peut se produire est signalé par le message [AMQ9465](#) ou [CSQX465I](#) sur les systèmes z/OS. Ce message indique que le problème doit être examiné.

## Remarques spéciales relatives à l'équilibrage de charge

Lorsque le cluster équilibre la charge entre plusieurs instances d'une file d'attente, les applications consommatrices doivent traiter les messages sur chacune des instances. Si une ou plusieurs de ces applications consommatrices arrêtent ou arrêtent de traiter des messages, il est possible que la mise en cluster continue à envoyer des messages à ces instances de la file d'attente. Dans ce cas, ces messages ne sont pas traités tant que les applications ne fonctionnent pas à nouveau correctement. Pour cette raison, la surveillance des applications est une partie importante de la solution et des mesures doivent être prises pour rediriger les messages dans cette situation. Un exemple de mécanisme permettant d'automatiser ce type de surveillance est disponible dans cet exemple: [Exemple de programme de surveillance de file d'attente de cluster \(AMQSCLM\)](#).

### Concepts associés

«Optimisation des réseaux de publication / abonnement distribués», à la page 389

Utilisez les conseils d'optimisation de cette section pour améliorer les performances de vos clusters et hiérarchies de publication / abonnement distribué IBM MQ .

«Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement», à la page 395

Un concept important dans les performances de messagerie asynchrone est *balance*. A moins que les consommateurs de messages ne soient équilibrés avec les expéditeurs de messages, il existe un risque qu'un arriéré de messages non consommés puisse s'accumuler et affecter sérieusement les performances de plusieurs applications.

## Surveillance du changement de file d'attente de transmission

Il est important de surveiller le processus de commutation des files d'attente de transmission par les canaux émetteurs de cluster afin de minimiser l'impact sur votre entreprise. Par exemple, vous ne devez pas tenter ce processus lorsque la charge de travail est élevée ou en permutant plusieurs canaux simultanément.

## Le processus de commutation des canaux

Le processus utilisé pour changer de canal est le suivant:

1. Le canal ouvre la nouvelle file d'attente de transmission pour l'entrée et commence à en extraire des messages (à l'aide de l'ID de corrélation)
2. Un processus d'arrière-plan est lancé par le gestionnaire de files d'attente pour déplacer les messages mis en file d'attente pour le canal de son ancienne file d'attente de transmission vers sa nouvelle file d'attente de transmission. Pendant que les messages sont déplacés, les nouveaux messages du canal sont mis en file d'attente dans l'ancienne file d'attente de transmission afin de préserver le séquençement. Ce processus peut prendre un certain temps s'il existe un grand nombre de messages pour le canal dans son ancienne file d'attente de transmission ou si de nouveaux messages arrivent rapidement.
3. Lorsqu'aucun message validé ou non validé ne reste en file d'attente pour le canal dans son ancienne file d'attente de transmission, le commutateur est terminé. Les nouveaux messages sont désormais placés directement dans la nouvelle file d'attente de transmission.

Pour éviter l'éventualité d'une commutation simultanée de nombreux canaux, IBM MQ permet de commuter la file d'attente de transmission d'un ou de plusieurs canaux qui ne sont pas en cours d'exécution. Sous :

- IBM MQ for Multiplatforms la commande est appelée **runswch1**
- IBM MQ for z/OS l'utilitaire CSQUTIL peut être utilisé pour traiter une commande SWITCH CHANNEL à la place

## Surveillance du statut des opérations de commutation

Pour comprendre le statut des opérations de commutation, les administrateurs peuvent effectuer les actions suivantes:

- Surveillez le journal des erreurs du gestionnaire de files d'attente (AMQERR01.LOG) où des messages sont générés pour indiquer les étapes suivantes au cours de l'opération:
  - L'opération de commutation a démarré
  - Le déplacement des messages a commencé
  - Mises à jour périodiques du nombre de messages restant à déplacer (si l'opération de commutation ne se termine pas rapidement)
  - Le déplacement des messages est terminé
  - L'opération de commutation est terminée

Sous z/OS, ces messages sont consignés dans le journal des travaux du gestionnaire de files d'attente, et non dans le journal des travaux de l'initiateur de canal, bien qu'un seul message soit consigné par un canal dans le journal des travaux de l'initiateur de canal s'il déclenche une commutation lors du démarrage.

- Utilisez la commande `DISPLAY CLUSQMGR` pour interroger la file d'attente de transmission actuellement utilisée par chaque canal émetteur de cluster.
- Exécutez la commande **runswch1** (ou `CSQUTIL` sous z/OS) en mode de requête pour vérifier l'état de commutation d'un ou de plusieurs canaux. La sortie de cette commande identifie les éléments suivants pour chaque canal:
  - Indique si le canal a une opération de commutation en attente
  - La file d'attente de transmission depuis et vers laquelle le canal bascule
  - Nombre de messages restant dans l'ancienne file d'attente de transmission

Chaque commande est vraiment utile, car dans un appel, vous pouvez déterminer le statut de chaque canal, l'impact d'un changement de configuration et si toutes les opérations de commutation sont terminées.

## Problèmes potentiels pouvant se produire

Voir [Problèmes potentiels lors de la commutation de files d'attente de transmission](#) pour obtenir la liste des problèmes qui peuvent se produire lors de la commutation de files d'attente de transmission, leurs causes et les solutions les plus probables.

### Concepts associés

«Optimisation des réseaux de publication / abonnement distribués», à la page 389

Utilisez les conseils d'optimisation de cette section pour améliorer les performances de vos clusters et hiérarchies de publication / abonnement distribué IBM MQ .

«Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement», à la page 395

Un concept important dans les performances de messagerie asynchrone est *balance*. A moins que les consommateurs de messages ne soient équilibrés avec les expéditeurs de messages, il existe un risque qu'un arriéré de messages non consommés puisse s'accumuler et affecter sérieusement les performances de plusieurs applications.

Multi

## Surveillance de l'équilibrage des applications

Vous pouvez utiliser la commande **DISPLAY APSTATUS** pour surveiller l'état de l'équilibrage des applications dans un cluster uniforme et pour déterminer pourquoi l'application n'est pas équilibrée si cela est inattendu.

### Surveillance de l'état en cours des applications dans les gestionnaires de files d'attente d'un cluster

A partir de n'importe quel gestionnaire de files d'attente dans un cluster uniforme, vous pouvez obtenir une vue d'ensemble de l'état en cours des applications dans tous les gestionnaires de files d'attente d'un cluster en exécutant la commande `DIS APSTATUS` .

A partir de IBM MQ 9.2.0 , la zone **TYPE** est également affichée dans la sortie.

Par exemple, après le démarrage d'un gestionnaire de files d'attente, vous pouvez voir une sortie similaire à la suivante:

```

1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                      MOVCOUNT(8)
BALANCED(UNKNOWN)
TYPE (APPL)

```

Cela montre qu'il existe une application appelée MYAPP dans le cluster uniforme, et qu'il existe actuellement huit instances, qui sont toutes considérées comme mobiles autour du cluster uniforme. La



valeur équilibrée de UNKNOWN est une valeur temporaire, qui indique que le gestionnaire de files d'attente n'a pas encore tenté de rééquilibrer l'application si nécessaire.

Après un court laps de temps, vous verrez probablement la sortie suivante:

```
1 : DIS APSTATUS(*) type(APPL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                CLUSTER(UNIDEMO)
COUNT(8)                       MOVCCOUNT(8)
BALANCED(NO)
TYPE (APPL)
```

Cette sortie indique que l'application comporte huit instances, mais qu'elles ne sont pas équilibrées dans le cluster uniforme. A ce stade, il est utile d'examiner la distribution des applications dans le cluster.

Pour ce faire, exécutez à nouveau la commande **DIS APSTATUS** . Notez que vous pouvez exécuter cette commande sur n'importe quel gestionnaire de files d'attente du cluster uniforme:

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(6)                       MOVCCOUNT(6)
BALSTATE(HIGH)                  LMSGDATE(2019-05-24)
LMSGTIME(13:11:10)              QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:03)              QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(1)                       MOVCCOUNT(1)
BALSTATE(LOW)                   LMSGDATE(2019-05-24)
LMSGTIME(13:11:07)              QMNAME(UNID003)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
```

A ce stade, vous pouvez voir que le gestionnaire de files d'attente UNID001 comporte six instances, mais que les gestionnaires de files d'attente UNID0002 et UNID0003 n'en comportent qu'une seule. La sortie BALSTATE donne une indication de l'état d'équilibrage lors de la dernière fois que le gestionnaire de files d'attente l'a signalé. Cependant, sachez que le nombre d'instances peut être plus récent que la zone BALSTATE.

Cette sortie donne également une bonne indication que le cluster uniforme distribue des informations sur cette application. L'application est connue de toutes les instances de gestionnaire de files d'attente du cluster uniforme et la date et l'heure du dernier message sont très récentes.

De plus, la zone ACTIVE indique que tous les gestionnaires de files d'attente du cluster sont considérés comme communiquant entre eux ; si ACTIVE est défini sur NO sur un gestionnaire de files d'attente, cela signifie que la communication régulière avec ce dernier a été interrompue.

S'il restait à se rééquilibrer, la sortie finirait par ressembler à ceci:

```
1 : DIS APSTATUS(*) type(QMGR)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                       MOVCCOUNT(3)
BALSTATE(OK)                    LMSGDATE(2019-05-24)
LMSGTIME(13:14:22)              QMNAME(UNID001)
QMID(UNID001_2019-05-24_13.09.35)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
COUNT(3)                       MOVCCOUNT(3)
BALSTATE(OK)                    LMSGDATE(2019-05-24)
LMSGTIME(13:13:53)              QMNAME(UNID002)
QMID(UNID002_2019-05-24_13.09.39)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)                ACTIVE(YES)
```

```

COUNT(2)
BALSTATE(OK)
LMSGTIME(13:13:47)
QMID(UNID003_2019-05-24_13.09.43)
TYPE (QMGR)
MOVCOUNT(2)
LMSGDATE(2019-05-24)
QMNAME(UNID003)

```

L'état BALSTATE des gestionnaires de files d'attente est désormais OK, ce qui indique qu'un état stabilisé a été atteint.

## Surveillance d'instances d'application individuelles

Vous pouvez examiner les instances d'application individuelles, mais vous pouvez le faire pour chaque gestionnaire de files d'attente. Voir la sortie sur UNID001:

```

1 : DIS APSTATUS(*) type(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02EA0A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1) IMMREASN(NONE)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E50A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1) IMMREASN(NONE)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(YES)
AMQ8932I: Display application status details.
APPLNAME(MYAPP)
CONNTAG(MQCT02DFE75C02E60A20UNID001_2019-05-24_13.09.35MYAPP)
CONNS(1) IMMREASN(NONE)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(YES)
TYPE (LOCAL)
One MQSC command read.

```

La présence de MOBILE (YES) indique que cette instance peut être déplacée si nécessaire vers un autre gestionnaire de files d'attente dans le cluster. Dans l'exemple suivant, l'instance d'application n'est pas déplaçable car elle ne se connecte pas en tant que client:

```

3 : DISPLAY APSTATUS('ServerApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ServerApp)
CONNTAG(MQCT02DFE75C01800B20UNID001_2019-05-24_13.09.35ServerApp)
CONNS(1) IMMREASN(NOTCLIENT)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

```

Vous pouvez utiliser la zone **CONNTAG** pour afficher les connexions de gestionnaire de files d'attente individuelles de cette instance, ce qui peut être utile si l'instance d'application est IMMREASN (NOTRECONN). Dans l'exemple suivant, l'application client n'est pas déplaçable et l'examen des options de connexion montre que MQCNO\_RECONNECT ne figure pas dans la zone **CONNOPTS**.

```

1 : DISPLAY APSTATUS('ClientApp') TYPE(LOCAL)
AMQ8932I: Display application status details.
APPLNAME(ClientApp)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)
CONNS(1) IMMREASN(NOTRECONN)
IMMCOUNT(0) IMMDATE( )
IMMTIME( ) MOVABLE(NO)
TYPE (LOCAL)

2 : DISPLAY CONN(*) TYPE(CONN) WHERE(CONNTAG eq
'MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp') ALL
AMQ8276I: Display Connection details.
CONN(02DFE75C01CB0B20)
EXTCONN(414D5143554E49443030312020202020)
TYPE (CONN)
PID(14656) TID(20)

```

APPLDESC(IBM MQ Channel)	APPLTAG(ClientApp)
APPLTYPE(USER)	ASTATE(NONE)
CHANNEL(SYSTEM.DEF.SVRCONN)	CLIENTID( )
CONNNAME(127.0.0.1)	
CONNOPTS(MQCNO_HANDLE_SHARE_BLOCK,MQCNO_SHARED_BINDING)	
USERID(MyUserId)	UOWLOG( )
UOWSTDA( )	UOWSTTI( )
UOWLOGDA( )	UOWLOGTI( )
URTYPE(QMGR)	
EXTURID(XA_FORMATID[] XA_GTRID[] XA_BQUAL[])	
QMURID(0,0)	UOWSTATE(NONE)
CONNTAG(MQCT02DFE75C01CB0B20UNID001_2019-05-24_13.09.35ClientApp)	
TYPE (CONN)	

### Concepts associés

[Equilibrage automatique des applications](#)

z/OS

## Monitoring performance and resource usage on z/OS

Use this topic to understand the facilities available to monitor the performance, and resource usage of your IBM MQ for z/OS subsystems.

### Related tasks

[Configuring queue managers on z/OS](#)

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

z/OS

## Introduction to monitoring IBM MQ for z/OS

Use this topic as an overview of the monitoring facilities available for IBM MQ for z/OS. For example, obtaining snapshots, using IBM MQ trace, online monitoring, and events.

This topic describes how to monitor the performance and resource usage of IBM MQ.

- It outlines some of the information that you can retrieve and briefly describes a general approach to investigating performance problems. See [“Investigating performance problems” on page 337](#) for more information.
- It describes how you can collect statistics about the performance of IBM MQ by using SMF records.
- It describes how to gather accounting data to enable you to charge your customers for their use of your IBM MQ systems.
- It describes how to use IBM MQ events (alerts) to monitor your systems.

Here are some of the tools you might use to monitor IBM MQ; they are described in the sections that follow:

- Tools provided by IBM MQ:
  - [Using DISPLAY commands](#)
  - [“Using CICS adapter statistics” on page 332](#)
  - [“Using IBM MQ events” on page 334](#)
- z/OS service aids:
  - [“Using System Management Facility” on page 335](#)
- Other IBM licensed programs:
  - [Using the Resource Measurement Facility](#)
  - [Using Tivoli Decision Support for z/OS](#)
  - [Using the CICS monitoring facility](#)

Information about interpreting the data gathered by the performance statistics trace is given in [“Interpreting IBM MQ for z/OS performance statistics” on page 338](#).

Information about interpreting the data gathered by the accounting trace is given in [“Interpreting IBM MQ for z/OS accounting data” on page 375](#).

## **Getting snapshots of IBM MQ using the DISPLAY commands**

IBM MQ provides the MQSC facility which can give a snapshot of the performance, and resource usage using the DISPLAY commands.

You can get an idea of the current state of IBM MQ by using the DISPLAY commands and, for the CICS adapter, the CICS adapter panels.

### **Using DISPLAY commands**

You can use the IBM MQ MQSC DISPLAY or PCF Inquire commands to obtain information about the current state of IBM MQ. They provide information about the status of the command server, process definitions, queues, the queue manager, and its associated components. These commands are:

<b>MQSC command</b>	<b>PCF command</b>
DISPLAY ARCHIVE	Inquire Archive
DISPLAY AUTHINFO	Inquire Authentication Information Object
DISPLAY CFSTATUS	Inquire CF Structure Status
DISPLAY CFSTRUCT	Inquire CF Structure
DISPLAY CHANNEL	Inquire Channel
DISPLAY CHINIT	Inquire Channel Initiator
DISPLAY CHSTATUS	Inquire Channel Status
DISPLAY CMDSERV	
DISPLAY CLUSQMGR	Inquire Cluster Queue Manager
DISPLAY CONN	Inquire Connection
DISPLAY GROUP	Inquire Group
DISPLAY LOG	Inquire Log
DISPLAY PROCESS	Inquire Process
DISPLAY QMGR	Inquire Queue Manager
DISPLAY QSTATUS	Inquire Queue Status
DISPLAY QUEUE	Inquire Queue
DISPLAY SECURITY	Inquire Security
DISPLAY STGCLASS	Inquire Storage Class
DISPLAY SYSTEM	Inquire System
DISPLAY TRACE	
DISPLAY USAGE	Inquire Usage

For the detailed syntax of each command, see MQSC commands or PCF commands. All of the functions of these commands (except DISPLAY CMDSERV and DISPLAY TRACE) are also available through the operations and control panels.

These commands provide a snapshot of the system only at the moment the command was processed. If you want to examine trends in the system, you must start an IBM MQ trace and analyze the results over a period of time.

## **Using CICS adapter statistics**

If you are an authorized CICS user, you can use the CICS adapter control panels to display CICS adapter statistics dynamically.

These statistics provide a snapshot of information related to CICS thread usage and situations when all threads are busy. The display connection panel can be refreshed by pressing the Enter key.

For more information about configuring the IBM MQ CICS adapter, see the [Configuring connections to MQ](#) section of the CICS documentation.

## **Using IBM MQ trace**

You can record performance statistics and accounting data for IBM MQ by using the IBM MQ trace facility. Use this topic to understand how to control IBM MQ trace.

The data generated by IBM MQ is sent to:

- The System Management Facility (SMF), specifically as SMF record type 115, subtypes 1 and 2 for the performance statistics trace
- The SMF, specifically as SMF record type 116, subtypes zero, 1, and 2 for the accounting trace.


If you prefer, the data generated by the IBM MQ accounting trace can also be sent to the generalized trace facility (GTF).

### **Starting IBM MQ trace**

You can start the IBM MQ trace facility at any time by issuing the IBM MQ [START TRACE](#) command.


Accounting data can be lost if the accounting trace is started or stopped while applications are running. To collect accounting data successfully, the following conditions must apply:

- The accounting trace must be active when an application starts, and it must still be active when the application finishes.
- If the accounting trace is stopped, any accounting data collection that was active stops.

You can also start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro.  These parameters are described in [Using CSQ6SYSP](#).

Before starting an IBM MQ trace, read [“Using System Management Facility”](#) on page 335.

### **Controlling IBM MQ trace**

To control the IBM MQ trace data collection at start-up, specify values for the parameters in the CSQ6SYSP macro when you customize IBM MQ.  See [Using CSQ6SYSP](#) for details.

You can control IBM MQ tracing when the queue manager is running with these commands:

- START TRACE
- ALTER TRACE
- STOP TRACE

You can choose the destination to which trace data is sent. Possible destinations are:

#### **SMF**

System Management Facility

#### **GTF**

Generalized Trace Facility (accounting trace only)

#### **SRV**

Serviceability routine for diagnostic use by IBM service personnel

For daily monitoring, information is sent to SMF (the default destination). SMF data sets typically contain information from other systems; this information is not available for reporting until the SMF data set is dumped.

You can also send accounting trace information to the GTF. This information has an event identifier of 5EE.

**z/OS** The The MQI call and user parameter, and z/OS generalized trace facility (GTF) describes how to deal with IBM MQ trace information sent to the GTF.

For information about IBM MQ commands, see MQSC commands.

## Effect of trace on IBM MQ performance

Using the IBM MQ trace facility can have a significant effect on IBM MQ and transaction performance. For example, if you start a global trace for class 1 or for all classes, it is likely to increase processor usage and transaction response times by approximately 50%. However, if you start a global trace for classes 2 - 4 alone, the increase in processor usage and transaction response times is likely to be less than 1% additional processor cost to the cost of IBM MQ calls. The same applies for a statistics or accounting trace.

## **z/OS** Using IBM MQ online monitoring

You can collect monitoring data for queues and channels (including automatically defined cluster-server channels) by setting the MONQ, MONCHL, and MONACLS attributes.

Table 33 on page 334 summarizes the commands to set these attributes at different levels and to display the monitoring information.

Attribute	Applicable at this level	Set using command	Display monitoring information using command
MONQ	Queue	DEFINE QLOCAL DEFINE QMODEL ALTER QLOCAL ALTER QMODEL	DISPLAY QSTATUS
	Queue manager	ALTER QMGR	
MONCHL	Channel	DEFINE CHANNEL ALTER CHANNEL	DISPLAY CHSTATUS
	Queue manager	ALTER QMGR	
MONACLS	Queue manager	ALTER QMGR	

For full details of these commands, see MQSC commands. For more information about online monitoring, see “Surveillance de votre réseau IBM MQ” on page 5.

## **z/OS** Using IBM MQ events

IBM MQ instrumentation events provide information about errors, warnings, and other significant occurrences in a queue manager. You can monitor the operation of all your queue managers by incorporating these events into your own system management application.

IBM MQ instrumentation events fall into the following categories:

### Queue manager events

These events are related to the definitions of resources within queue managers. For example, an application attempts to put a message to a queue that does not exist.

### Performance events

These events are notifications that a threshold condition has been reached by a resource. For example, a queue depth limit has been reached, or the queue was not serviced within a predefined time limit.

### Channel events

These events are reported by channels as a result of conditions detected during their operation. For example, a channel instance is stopped.

### Configuration events

These events are notifications that an object has been created, changed, or deleted.

When an event occurs, the queue manager puts an *event message* on the appropriate *event queue*, if defined. The event message contains information about the event that can be retrieved by a suitable IBM MQ application.

IBM MQ events can be enabled using the IBM MQ commands or the operations and control panels.

See “Types d'événements” on page 12 for information about the IBM MQ events that generate messages, and for information about the format of these messages. See [Event message reference](#) for information about enabling the events.

## Using System Management Facility

You can use SMF to collect statistics and accounting information. To use SMF, certain parameters must be set in z/OS and in IBM MQ.

System management facility (SMF) is a z/OS service aid used to collect information from various z/OS subsystems. This information is dumped and reported periodically, for example, hourly. You can use SMF with the IBM MQ trace facility to collect data from IBM MQ. In this way you can monitor *trends*, for example, in system utilization and performance, and collect accounting information about each user ID using IBM MQ.

To record performance statistics (record type 115) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(115))
```

To record accounting information (record type 116) to SMF specify the following in the SMFPRMxx member of SYS1.PARMLIB or with the SETSMF z/OS operator command.

```
SYS(TYPE(116))
```

To use the z/OS command SETSMF, either PROMPT(ALL) or PROMPT(LIST) must be specified in the SMFPRM xx member. See [SMFPRMxx \(system management facilities \(SMF\) parameters\)](#) for more information.

You can start collecting some trace information automatically if you specify YES on the SMFSTAT (SMF STATISTICS) and SMFACCT (SMF ACCOUNTING) parameters of the CSQ6SYSP macro; this is described in [Using CSQ6SYSP](#).

Specifying YES on the SMFSTAT and SMFACCT parameters enables you to collect trace information as a queue manager starts.

You can also start collection of the data when the queue manager is running with the **START TRACE** command, specifying START TRACE(A) or START TRACE(S).

You can turn on or off the recording of accounting information at the queue or queue manager level using the ACCTQ parameter of the **DEFINE QLOCAL**, **DEFINE QMODEL**, **ALTER QLOCAL**, **ALTER QMODEL**, or **ALTER QMGR** commands. See [MQSC commands](#) for details of these commands.

You can control the collection of channel accounting data at the channel or queue manager level using the **STATCHL** parameter of the **DEFINE CHANNEL**, **ALTER CHANNEL** or **ALTER QMGR** commands.

You can specify the interval at which IBM MQ collects statistics and accounting data in one of these ways:

- You can collect statistics data and accounting data at different intervals, using STATIME (statistics data) and ACCTIME (accounting data) in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data at the same interval by specifying a value for STATIME in your system parameters (described in [Using CSQ6SYSP](#)).
- You can collect statistics data and accounting data by specifying zero for STATIME.

SMF must be running before you can send data to it. For more information about SMF, see the [z/OS MVS System Management Facilities \(SMF\)](#) manual.

For the statistics and accounting data to be reset, at least one MQI call must be issued during the accounting interval.

## Allocating additional SMF buffers

When you start a trace, you must ensure that you allocate adequate SMF buffers. Specify SMF buffering on the VSAM BUFSP parameter of the access method services DEFINE CLUSTER statement. Specify CISZ(4096) and BUFSP(81920) on the **DEFINE CLUSTER** statement for each SMF VSAM data set.

If an SMF buffer shortage occurs, SMF rejects any trace records sent to it. IBM MQ sends a CSQW133I message to the z/OS console when this occurs. IBM MQ treats the error as temporary and remains active even though SMF data can be lost. When the shortage has been alleviated and trace recording has resumed, IBM MQ sends a CSQW123I message to the z/OS console.

## Reporting data in SMF

You can use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed.

There are several ways to report on this data, for example:

- Write an application program to read and report information from the SMF data set. You can then tailor the report to fit your exact needs.
- Use Performance Reporter to process the records. For more information, see [“Using other products with IBM MQ”](#) on page 337.

## No interval CLASS(03) SMF accounting records produced during long running processes

You are collecting CLASS(3) SMF116 accounting records for IBM MQ, but are getting no records produced while a long running process runs.

The CLASS(3) SMF116 accounting records normally are produced only when a process ends. For long running processes, for example CICS, this might not produce a sufficient number of records, as the process can run for a month or longer. However, you might want to gather SMF116 records at set time intervals while a process is running.

To gather CLASS(3) SMF116 accounting records you must set the following:

### SMFACCT

=YES

### SMFSTAT

=YES or NO, where

#### YES

Causes records to be produced if a collection broadcast is received.

#### No

Causes you to get a CLASS(3) SMF116 record produced only when a process ends



and issue the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(03)
```

If you have set SMFSTAT=YES and a collection broadcast occurs, an interval CLASS(3) SMF116 accounting record is produced for any process currently running that was also running at the time of the previous collection broadcast.

You can set the collection broadcast to occur on a regular time interval by setting STATIME in [CSQ6SYSP](#) as follows:

- If your STATIME has been set to a value greater than 0, that is your broadcast interval in minutes.
- If your STATIME = 0 the SMF broadcast of your system is used (SMF INTVAL)
- If your STATIME = 0 and your SMF INTVAL is not set, no broadcast occurs and no interval records are produced

### **Using other products with IBM MQ**

You can use other products to help you to improve the presentation of, or to augment statistics related to, performance and accounting. For example, Resource Measurement Facility, Tivoli Decision Support, and CICS monitoring.

### **Using the Resource Measurement Facility**

Resource Measurement Facility (RMF) is an IBM licensed program (program number 5685-029) that provides system-wide information about processor utilization, I/O activity, storage, and paging. You can use RMF to monitor the utilization of physical resources across the whole system dynamically. For more information, see the [z/OS Resource Measurement Facility User's Guide](#).

### **Using Tivoli Decision Support for z/OS**

You can use Tivoli Decision Support for z/OS to interpret RMF and SMF records.

Tivoli Decision Support for z/OS is an IBM licensed program (program number 5698-B06) that enables you to manage the performance of your system by collecting performance data in a Db2® database and presenting the data in various formats for use in systems management. Tivoli Decision Support for can generate graphic and tabular reports using systems management data it stores in its Db2 database. It includes an administration dialog, a reporting dialog, and a log collector, all of which interact with a standard Db2 database.

This is described in the [IBM Tivoli Decision Support for z/OS: Administration Guide and Reference](#).

### **Using the CICS monitoring facility**

The CICS monitoring facility provides performance information about each CICS transaction running. It can be used to investigate the resources used and the time spent processing transactions. For background information, see the [CICS Performance Guide](#) and [Developing CICS System Programs](#), together with the two companion reference manuals, formerly called the [CICS Customization Guide](#).

### **Investigating performance problems**

Performance problems can arise from various factors. For example, incorrect resource allocation, poor application design, and I/O restraints. Use this topic to investigate some of the possible causes of performance problems.

Performance can be adversely affected by:

- Buffer pools that are an incorrect size
- Lack of real storage
- I/O contention for page sets or logs

- Log buffer thresholds that are set incorrectly
- Incorrect setting of the number of log buffers
- Large messages
- Units of recovery that last a long time, incorporating many messages for each sync point
- Messages that remain on a queue for a long time
- RACF® auditing
- Unnecessary security checks
- Inefficient program design

When you analyze performance data, always start by looking at the overall system before you decide that you have a specific IBM MQ problem. Remember that almost all symptoms of reduced performance are magnified when there is contention. For example, if there is contention for DASD, transaction response times can increase. Also, the more transactions there are in the system, the greater the processor usage and greater the demand for both virtual and real storage.

In such situations, the system shows heavy use of *all* its resources. However, the system is actually experiencing normal system stress, and this stress might be hiding the cause of a performance reduction. To find the cause of such a loss of performance, you must consider all items that might be affecting your active tasks.

## Investigating the overall system

Within IBM MQ, the performance problem is either increased response time or an unexpected and unexplained heavy use of resources. First check factors such as total processor usage, DASD activity, and paging. An IBM tool for checking total processor usage is resource management facility (RMF). In general, you must look at the system in some detail to see why tasks are progressing slowly, or why a specific resource is being heavily used.


Start by looking at general task activity, then focus on particular activities, such as specific tasks or a specific time interval.

Another possibility is that the system has limited real storage; therefore, because of paging interrupts, the tasks progress more slowly than expected.

## Investigating individual tasks

You can use the accounting trace to gather information about IBM MQ tasks. These trace records tell you a great deal about the activity that the task has performed, and about how much time the task spent suspended, waiting for latches. The trace record also includes information about how much Db2 and coupling facility activity were performed by the task.

Interpreting IBM MQ accounting data is described in [“Interpreting IBM MQ for z/OS accounting data” on page 375](#).

Long running units of work can be identified by the presence of message CSQR026I in the job log. This message indicates that a task has existed for more than three queue manager checkpoints and its log records have been shunted.  For a description of log record shunting, see [The log files](#).

## Interpreting IBM MQ for z/OS performance statistics

Use this topic as an index to the different SMF records created by IBM MQ for z/OS.

IBM MQ for z/OS performance statistics are written as SMF type 115 records. Statistics records are produced periodically at a time interval specified by the **STATIME** parameter of the CSQ6SYSP system parameter module, or at the SMF global recording interval if you specify zero for **STATIME**. The information provided in the SMF records comes from the following components of IBM MQ:

<b>Buffer manager</b>	Manages the buffer pools in virtual storage and the writing of pages to page sets as the buffer pools become full. Also manages the reading of pages from page sets.
<b>Coupling facility manager</b>	Manages the interface with the coupling facility.
<b>Data manager</b>	Manages the links between messages and queues. It calls the buffer manager to process the pages with messages on them.
<b>Db2 manager</b>	Manages the interface with the Db2 database that is used as the shared repository.
<b>Lock manager</b>	Manages locks
<b>Log manager</b>	Manages the writing of log records, which are essential for maintaining the integrity of the system if there is a back out request, or for recovery, if there is a system or media failure.
<b>Message manager</b>	Processes all IBM MQ API requests.
<b>Storage manager</b>	Manages storage, for example, storage pool allocation, expansion, and deallocation.
<b>Topic manager</b>	Manages the topic and subscription information
<b>Coupling facility SMDS manager</b>	Manages the shared message data sets (SMDS) for large messages stored in the coupling facility.

IBM MQ statistics are written to SMF as SMF type 115 records. The following subtypes can be present:

**1**

System information, for example, related to the logs and storage.

**2**

Information about number of messages and paging information. Queue sharing group information related to the coupling facility and Db2.

**5 and 6**

Detailed information about internal storage usage in the queue manager address space. While you can view this information, some of it is intended only for IBM use.

**7**

Storage manager summary information. While you can view this information, some of it is intended only for IBM use.

**201**

Page set input/output information

**215**

Buffer pool information

**216**

Queue information

**231**

System information for the channel initiator address space.

Note that:

- Subtype 1, 2, 201, and 215 records are created with statistics trace class 1.
- Subtype 5, 6, and 7 records are created with statistics trace class 3.
- Subtype 231 records are created with statistics trace class 4.
- Subtype 216 records are created with statistics trace class 5.

The subtype is specified in the SM115STF field (shown in [Table 34 on page 343](#) ).

## **Commandes requises pour collecter les enregistrements SMF de type 115 et 116**

Utilisez cette rubrique comme référence pour les commandes nécessaires à la collecte des enregistrements SMF de type 115 et 116.

### Utilisation de la commande **DÉMARRER TRACE**

Utilisez la version dynamique du [COMMENCER LE TRAÇAGE](#) commande avec les options suivantes pour collecter les enregistrements :

- `START TRACE(STAT) DEST SMF CLASS(*)` et `START TRACE(ACCTG) DEST SMF CLASS(*)` démarre les traces pour les classes 1 à 3
- `START TRACE(STAT) DEST SMF CLASS(4)` et `START TRACE(ACCTG) DEST SMF CLASS(4)` démarre respectivement les statistiques de l'initiateur du canal et les données de comptabilité du canal.

Voir [Planification des données SMF de l'initiateur de canal](#) pour plus de détails sur les informations sur l'initiateur de canal de classe 4.

- `START TRACE(STAT) DEST SMF CLASS(5)` démarre les statistiques de file d'attente.



**Avertissement :** Si vous utilisez la version console de la commande `addcpf` au début de la commande que vous émettez.

Pour les statistiques du canal `CLASS(4)`, vous devez définir l'attribut `STATCHL` sur la définition `CHANNEL`. Pour plus d'informations, voir [«Contrôle de la collecte des statistiques de canal»](#), à la page 151.

Pour les statistiques de file d'attente `CLASS(5)`, vous devez définir l'attribut `STATQ` sur la définition `QUEUE` et/ou `QMGR`. Pour plus d'informations, voir [«Contrôle de la collecte des statistiques de file d'»](#), à la page 150.

### Enregistrements SMF type 115 -IBM MQ statistiques

Pour vérifier la collecte de statistiques en cours, émettez le [AFFICHER LA TRACE](#) commande `DISPLAY TRACE(STAT)` . Les informations suivantes s'affichent :

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
02 STAT 01,02,03,04 SMF * *
END OF TRACE REPORT
```

**Remarque :** La définition de l'attribut `ZPARAM SMFSTAT` ou `ZPARAM SMFACCT` sur un astérisque contrôle désormais uniquement les classes 1, 2 et 3. Vous devez activer les classes 4 et 5 à l'aide de la commande `START TRACE`.

### Données SMF type 116 classes 3 et 4 –IBM MQ enregistrements comptables des tâches et des canaux

Pour vérifier la collecte de statistiques en cours, émettez le [AFFICHER LA TRACE](#) commande `DISPLAY TRACE(ACCT)` . Les informations suivantes s'affichent :

```
RESPONSE=MPX1
CSQW127I QML1 CURRENT TRACE ACTIVITY IS -
TNO TYPE CLASS DEST USERID RMID
03 ACCTG * SMF * *
END OF TRACE REPORT
CSQ9022I QML1 CSQWVC1 'DISPLAY TRACE' NORMAL COMPLETION
```

Tu peux tourner SMF116 activation et désactivation dynamique de la collecte de données car aucun cycle du gestionnaire de files d'attente n'est nécessaire, à l'aide des commandes suivantes :

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
START TRACE(ACCTG) DEST(SMF) CLASS(4)

STOP TRACE(ACCTG) DEST(SMF) CLASS(3)
STOP TRACE(ACCTG) DEST(SMF) CLASS(4)
```

Les données de comptabilité de canal nécessitent une configuration supplémentaire pour produire le SMF. Les étapes sont les suivantes :

- Exécutez la commande pour activer la collecte des informations pour toutes les connexions expéditeur-récepteur et client :

```
ALTER QMGR STATCHL(HIGH)
```

- Exécutez la commande pour activer la collecte des informations pour les canaux de cluster définis automatiquement si le clustering est utilisé :

```
ALTER QMGR STATCLS(HIGH)
```

**Remarque :** Dans les instructions précédentes, les valeurs HIGH, MED ou LOW ont le même effet.

- Vérifiez que tous les canaux ont STATCHL défini sur QMGR :

```
DISPLAY CHANNEL(*) STATCHL
```

### Vérification du STATIME



**Avertissement :** Vous devez définir l'intervalle sur cinq minutes maximum pour vous conformer aux derniers outils, en utilisant le RÉGLER LE SYSTÈME commande, bien que la valeur par défaut soit de 30 minutes.

Si vous évaluez un groupe de partage de files d'attente (QSG), assurez-vous que STATIME est cohérent dans l'ensemble du QSG.

- Affichez les paramètres du système en utilisant les éléments suivants SYSTÈME D'AFFICHAGE commande:

```
DISPLAY SYSTEM
```

- Si la valeur STATIME est zéro, ce qui signifie que l'intervalle de statistiques est défini sur la valeur LPAR par défaut, cela est généralement OK.

Il y a eu des cas de volumes LPAR très élevés créant des durées variables de production de données SMF. Si lors de votre évaluation des données, les durées varient considérablement, définissez STATIME sur une valeur différente de zéro pour tous les gestionnaires de files d'attente.

- Les exemples suivants définissent l'intervalle STATIME sur cinq minutes :

```
SET SYSTEM STATIME(05)
```

ou si à IBM MQ 9.2.4 ou au-dessus:

```
SET SYSTEM STATIME(05.00)
```

**Remarque :** Le nouvel intervalle ne prend effet qu'une fois l'intervalle actuel écoulé. Vous devez donc effectuer la modification avant le début de la collecte des données.

**Vérification de la valeur ACCTIME pour les gestionnaires de files d'attente au niveau de la version IBM MQ 9.2.4 ou plus:**

- Si ACCTIME est défini sur -1, c'est la même chose que l'attribut STATIME.
- Si cette évaluation concerne un QSG, assurez-vous que l'ACCTIME est cohérent dans l'ensemble du QSG. Si ce n'est pas le cas, définissez les valeurs sur le même intervalle avant de commencer à collecter les données.
- Si ACCTIME est réglé sur 30 ou plus, modifiez-le pour qu'il ne dépasse pas 15 à l'aide du bouton RÉGLER LE SYSTÈME commande. Les exemples suivants définissent l'intervalle ACCTIME sur cinq minutes :

```
SET SYSTEM ACCTIME(05.00)
```

**Remarque :** Le nouvel intervalle ne prend effet qu'une fois l'intervalle actuel écoulé. Vous devez donc effectuer la modification avant le début de la collecte des données.

### Vérification du paramètre ACCTQ sur les gestionnaires de files d'attente

- Affichez le paramètre ACCTQ pour les gestionnaires de files d'attente que vous étudiez à l'aide de la commande suivante :

```
DISPLAY QMGR ACCTQ
```

- Si la valeur est ACCTQ(ON), aucune autre action n'est nécessaire. Sinon, lancez la commande suivante :

```
ALTER QMGR ACCTQ(ON)
```

- Si le clustering est utilisé, assurez-vous que la comptabilité est activée pour leSYSTEM.CLUSTER.TRANSMIT.QUEUE et toute autre file d'attente de transmission de cluster nommée hébergée par le gestionnaire de files d'attente. Pour déterminer si la comptabilité est activée pour les files d'attente de transmission du cluster :

- Affichez le paramètre ACCTQ pour la ou les files d'attente à l'aide de la commande suivante :

```
DISPLAY QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ
```

- Si la valeur est ACCTQ(ON), aucune autre action n'est nécessaire. Sinon, lancez la commande suivante :

```
ALTER QL(SYSTEM.CLUSTER.TRANSMIT.QUEUE) ACCTQ(ON)
```

- Pour toutes les autres files d'attente à volume élevé, vérifiez que la valeur de ACCTQ est ON ou définissez ACCTG sur QMGR.

### **Layout of an SMF type 115 record**

You can use this section as a reference for the format of an SMF type 115 record.

The standard layout for SMF records involves three parts:

#### **SMF header**

Provides format, identification, and time and date information about the record itself.

#### **Self-defining section**

Defines the location and size of the individual data records within the SMF record.

#### **Data records**

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

#### **Related reference**

[“The SMF header” on page 343](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 343](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

“Examples of SMF statistics records” on page 344  
 Use this topic to understand some example SMF records.

### **The SMF header**

Use this topic as a reference for the format of the SMF header.

Table 34 on page 343 shows the format of SMF record header (SM115).

*Table 34. SMF record 115 header description*

Offset: Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM115	SMF record header.	
0	0	Integer	2	SM115LEN	SMF record length.	14A0
2	2		2		Reserved.	
4	4	Integer	1	SM115FLG	System indicator.	5E
5	5	Integer	1	SM115RTY	Record type. The SMF record type, for IBM MQ statistics records this is always 115 (X'73').	73
6	6	Integer	4	SM115TME	Time when SMF moved record.	00355575
10	A	Integer	4	SM115DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM115SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM115SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM115STF	Record subtype.	0002
24	18	Character	3	SM115REL	IBM MQ version.	F6F0F0 (600)
27	1B		1		Reserved	
28	1C	Character	0	SM115END	End of SMF header and start of self-defining section.	

### **Self-defining sections**

Use this topic as a reference for format of the self-defining sections of the SMF record.

A self-defining section of a type 115 SMF record tells you where to find a statistics record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at fixed offsets from the start of the SMF record. Each statistics record can be identified by an eye-catcher string.

The following types of self-defining section are available to users for type 115 records. Each self-defining section points to statistics data related to one of the IBM MQ components. Table 35 on page 344 summarizes the sources of the statistics, the eye-catcher strings, and the offsets of the self-defining sections from the start of the SMF record header.

Table 35. Offsets to self-defining sections

Source of statistics	Record subtype (SM115STF)	Offset of self-defining section		Eye-catcher of data
		Dec	Hex	
Storage manager	1	100	X'64'	QSST
Log manager	1	116	X'74'	QJST
Message manager	2	36	X'24'	QMST
Data manager	2	44	X'2C'	QIST
No longer used. The self-defining section will be binary zeros.	2	52	X'34'	
Lock manager	2	60	X'3C'	QLST
Db2 manager	2	68	X'44'	Q5ST
Coupling Facility manager	2	76	X'4C'	QEST
Topic manager	2	84	X'54'	QTST
SMDS usage	2	92	X'5C'	QESD
Buffer manager - one for each buffer pool	215	36	X'24'	QPST
Channel initiator	231			QWSX
Data manager page set - one for each page set	201	36	X'24'	QIS1
Storage manager	5	36	X'24'	QSPH
Storage manager	6	36	X'24'	QSGM
Storage manager	7	36	X'24'	QSRS
Queues	216	36	X'24'	QQST

**Note:** Some of the storage manager information in subtype 5, 6 and 7 records is intended only for IBM use. Other self-defining sections that are not listed contain data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssllllnnn
```

where:

- *sssssss* is a fullword containing the offset from the start of the SMF record.
- *llll* is a halfword giving the length of this data record.
- *nnn* is a halfword giving the number of data records in this SMF record.

For more information see, “Examples of SMF statistics records” on page 344.

**Note:** Always use offsets in the self-defining sections to locate the statistics records.

### Examples of SMF statistics records

Use this topic to understand some example SMF records.



Figure 20 on page 345 shows an example of part of the SMF record for subtype 1. Subtype 1 includes the storage manager and log manager statistics records. The SMF record header is shown underlined.

The self-defining section at offset X'64' refers to storage manager statistics and the self-defining section at offset X'74' refers to log manager statistics, both shown in **bold**.

The storage manager statistics record is located at offset X'0000011C' from the start of the header and is X'48' bytes long. There is one set of storage manager statistics, identified by the eye-catcher string QSST. The start of this statistics record is also shown in the example.

The log manager statistics record is located at offset X'00000164' from the start of the header and is X'78' bytes long. There is one set of log manager statistics, identified by the eye-catcher string QJST.

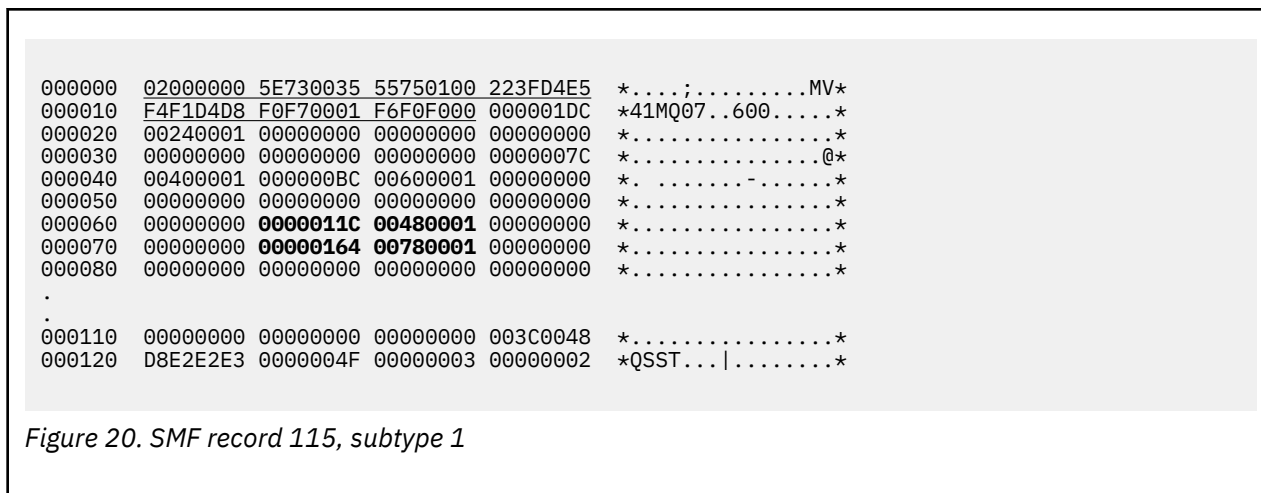


Figure 21 on page 346 shows an example of part of the SMF record for subtype 2. Subtype 2 includes the statistics records for the message, data, lock, coupling facility, topic, and Db2 managers. The SMF record header is shown underlined; the self-defining sections are shown alternately **bold** and *italic*.

- The self-defining section at offset X'24' refers to message manager statistics. The message manager statistics record is located at offset X'00000064' from the start of the header and is X'48' bytes long. There is one set of these statistics, identified by the eye-catcher string QMST.
- The self-defining section at offset X'2C' refers to data manager statistics. The data manager statistics record is located at offset X'000000AC' from the start of the header and is X'50' bytes long. There is one set of these statistics, identified by the eye-catcher string QIST.
- In releases prior to IBM MQ 9.1.0, the self-defining section at offset X'34' referred to buffer manager statistics. As this SMF record was taken from an IBM MQ 9.1.0 queue manager, the buffer manager self-defining section is set to zeros to indicate that there are no buffer manager statistics. Instead, these statistics are in SMF 115 subtype 215 records.
- The self-defining section at offset X'3C' refers to lock manager statistics. The lock manager statistics record is located at offset X'000000FC' from the start of the header and is X'20' bytes long. There is one set of these statistics, identified by the eye-catcher string QLST.
- The self-defining section at offset X'44' refers to Db2 manager statistics. The Db2 manager statistics record is located at offset X'0000011C' from the start of the header and is X'2A0' bytes long. There is one set of these statistics, identified by the eye-catcher string Q5ST.
- The self-defining section at offset X'4C' refers to coupling facility manager statistics. The coupling facility manager statistics record is located at offset X'000003BC' from the start of the header and is X'1008' bytes long. There is one set of these statistics, identified by the eye-catcher string QEST.
- The self-defining section at offset X'54' refers to topic manager statistics. The topic manager statistics record is located at offset X'000013C4' from the start of the header and is X'64' bytes long. There is one set of these statistics, identified by the eye-catcher string QTST.
- The self-defining section at offset X'5C' is for SMDS statistics. This self defining section is set to zeros indicating that SMDS is not being used.

```

000000 09F40000 5E730033 4DBE0113 142FD4E5 *.4..;...(. ....MV*
000010 F4F1D4D8 F2F10002 F9F1F000 00001428 *41MQ21..910.....*
000020 00240001 00000064 00480001 000000AC *.....*
000030 00500001 00000000 00000000 000000FC *.....*
000040 00200001 0000011C 02A00001 000003BC *.....*
000050 10080001 000013C4 00640001 00000000 *.....D.....*
000060 00000000 D40F0048 D8D4E2E3 00000000 *...M...QMST...*
000080 00000000 00000000 00000000 00000000 *.....*
000090 00000000 00000000 00000000 00000000 *.....*
0000A0 00000000 00000000 00000000 C90F0050 *.....I..&*
0000B0 D8C9E2E3 00000000 00000000 00000000 *QIST.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 D30F0020 *.....L...*
000100 D8D3E2E3 00000000 00000000 00000000 *QLST.....*
000110 00000000 00000000 00000000 F50F02A0 *.....5...*
000120 D8F5E2E3 00000008 00000000 00000000 *Q5ST.....*
.

```

Figure 21. SMF record 115, subtype 2

## Processing type 115 SMF records

Use this topic as a reference for processing type 115 SMF records.

You must process any data you collect from SMF to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the values of the following fields:

- SM115RTY, the SMF record number, must be X'73' (115)
- SM115STF, the record subtype, must be 0001, 0002, 0005, 0006, 0007, 0201, 0215, or 0231

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMFDL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility”](#) on page 335.

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

## Storage manager data records

Use this topic as a reference for storage manager data records.

The format of the storage manager statistics record is described in assembler macro thlqual.SCSQMACS(CSQDQSST).

The data contains information about the number of fixed and variable storage pools that the queue manager has allocated, expanded, contracted, and deleted during the statistics interval, plus the number of GETMAIN, FREEMAIN, and STORAGE requests to z/OS, including a count of those requests that were unsuccessful. Additional information includes a count of the number of times the short-on-storage condition was detected and a count of the number of abends that occurred as a result of that condition.

Additional data about storage usage in the queue manager is produced by class 2 and class 3 statistics trace. While you can view this information, some of it is intended only for IBM use.

- The format of the storage manager pool header statistics record, which is present in subtype 5 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSPH).
- The format of the storage manager getmain statistics record, which is present in subtype 6 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSGM).

- The format of the storage manager region summary record, which is present in subtype 7 records, is described in assembler macro thlqual.SCSQMACS(CSQDQSR).

## **Enregistrements de données du gestionnaire de journaux**

Utilisez cette rubrique comme référence pour le format des enregistrements de données du gestionnaire de journaux.

Le format de l'enregistrement des statistiques du gestionnaire de journaux est décrit dans la macro assembleur thlqual.SCSQMACS(CSQDQJST).

Dans les statistiques, ces comptages sont importants:

1. Nombre total de demandes d'écriture de journal:

$$N_{\text{logwrite}} = \text{QJSTWRNW} + \text{QJSTWRF}$$

2. Nombre total de demandes de lecture de journal:

$$N_{\text{logread}} = \text{QJSTRBUF} + \text{QJSTRACT} + \text{QJSTRARH}$$

Les symptômes de l'incident qui peuvent être examinés à l'aide des statistiques du gestionnaire de journaux sont décrits dans le tableau suivant.

### **Symptôme 1**

QJSTWTB est différent de zéro.

#### **Motif**

Les tâches sont interrompues lors de l'écriture de la mémoire tampon dans le journal actif.

Il peut y avoir des problèmes d'écriture dans le journal actif.

Le paramètre OUTBUFF dans CSQ6LOGP est trop petit.

#### **Action**

Recherchez les problèmes d'écriture dans le journal actif.

Augmentez la valeur du paramètre OUTBUFF dans CSQ6LOGP.

### **Symptôme 2**

Le rapport:  $\text{QJSTWTL} / N_{\text{logread}}$  est supérieur à 1%.

#### **Motif**

Des lectures de journal ont été initiées et ont dû être lues à partir d'un journal d'archivage, mais IBM MQ n'a pas pu allouer de fichier car des fichiers MAXRTU étaient déjà alloués.

#### **Action**

Augmentez MAXRTU.

**Symptôme 3**

Le rapport: QJSTRARH/N `logread` est supérieur à la normale.

**Motif**

La plupart des demandes de lecture de journal doivent provenir de la mémoire tampon de sortie ou du journal actif. Pour répondre aux demandes d'archivage, les enregistrements d'unité de récupération sont lus à partir de la mémoire tampon en mémoire, du journal actif et des journaux archivés.

Une unité de récupération de longue durée, s'étendant sur une période de plusieurs minutes, peut comporter des enregistrements de journal répartis sur plusieurs journaux différents. Cela dégrade les performances car un travail supplémentaire doit être effectué pour récupérer les enregistrements de journal.

**Action**

Modifiez l'application pour réduire la longueur d'une unité de récupération. Pensez également à augmenter la taille du journal actif afin de réduire la possibilité qu'une seule unité de récupération soit répartie sur plusieurs journaux.

**Autres pointeurs**

Le rapport N `logread` /N `logwrite` donne une indication de la quantité de travail à sauvegarder.

**Symptôme 4**

QJSTLLCP est supérieur à 10 par heure.

**Motif**

Sur un système occupé, vous devriez voir généralement 10 points de contrôle par heure. Si la valeur QJSTLLCP est supérieure à cette valeur, cela indique un problème dans la configuration du gestionnaire de files d'attente.

La raison la plus probable est que le paramètre LOGLOAD dans CSQ6SYSP est trop petit. L'autre événement qui provoque un point de contrôle est le moment où un journal actif se remplit et passe au fichier journal actif suivant. Si vos journaux sont trop petits, cela peut entraîner des points de contrôle fréquents.

QJSTLLCP est le nombre total de points de contrôle.

**Action**

Augmentez le paramètre LOGLOAD ou augmentez la taille de vos fichiers journaux selon vos besoins.

**Symptôme 5**

QJSTCmpFail > 0 ou QJSTCmpComp pas beaucoup moins que QJSTCmpUncmp

**Motif**

Le gestionnaire de files d'attente ne parvient pas à compresser de manière significative les enregistrements de journal.

QJSTCmpFail est le nombre de fois où le gestionnaire de files d'attente n'est pas parvenu à réduire la longueur d'enregistrement. Vous devez comparer le nombre à QJSTCmpReq (nombre de demandes de compression) pour voir si le nombre d'échecs est significatif.

QJSTCmpComp est le nombre total d'octets compressés écrits dans le journal et QJSTCmpUncmp est le nombre total d'octets avant compression. Aucun total ne contient d'octets écrits pour les enregistrements de journal qui ne pouvaient pas être compressés. Si les nombres sont similaires, la compression a peu d'avantages.

**Action**

Désactivez la compression de journal. Exécutez la commande SET LOG COMPLOG (NONE). Pour plus de détails, voir la commande [SET LOG](#).

**Remarque :** Dans le premier ensemble de statistiques généré après le démarrage du système, il peut y avoir une activité de journal importante en raison de la résolution des unités de récupération en cours.

## Modifications apportées à l'écriture zHyper

V 9.4.0

A partir de IBM MQ 9.3.5, le traitement d'écriture zHyper modifie le comportement des statistiques QJSTHWC et QJSTHWE.

Avant IBM MQ 9.3.5 :

- QJSTHWC est le nombre de fichiers journaux utilisés dans l'intervalle SMF qui se trouvent sur les volumes compatibles avec l'écriture zHyper. La fonction d'écriture zHyper d'un volume est obtenue au démarrage du gestionnaire de files d'attente et peut changer au fil du temps, de sorte que ces informations peuvent être périmées.
- QJSTHWE est le nombre de fichiers journaux utilisés dans l'intervalle SMF qui sont écrits avec l'option d'écriture zHyper activée. Cela se produit si le fichier journal se trouve sur un volume compatible avec l'écriture zHyper et si le gestionnaire de files d'attente a été activé pour l'écriture zHyper en définissant ZHYWRITE (YES).

Depuis IBM MQ 9.3.5:

- QJSTHWC est le nombre de fichiers journaux utilisés dans l'intervalle SMF qui se trouvent sur les volumes compatibles avec l'écriture zHyper. La fonction d'écriture zHyper d'un volume est obtenue au démarrage du gestionnaire de files d'attente et peut changer au fil du temps, de sorte que ces informations peuvent être périmées.
- QJSTHWE est le nombre de fichiers journaux utilisés dans l'intervalle SMF qui sont écrits avec l'option d'écriture zHyper activée. Cela se produit si le gestionnaire de files d'attente a été activé pour zHyperWrite en définissant ZHYWRITE (YES).



**Avertissement :** La capacité d'écriture zHyper d'un volume peut varier dans le temps. Depuis IBM MQ 9.3.5, cela peut entraîner des scénarios dans lesquels QJSTHWE est supérieur à QJSTHWC.

## Statistiques de lien zHyper

V 9.4.0

A partir de IBM MQ 9.4.0, les statistiques du gestionnaire de journaux ont été ajoutées pour suivre les performances de zHyperLink.

Les statistiques suivantes ont été ajoutées au QJST:

Nom	Tapez	Octets de longueur	Description
QJSTHLSCIW	Entier non signé	4	Nombre d'écritures de page unique qui ont demandé le lien zHyper.
QJSTHLMCIW	Entier non signé	4	Nombre d'écritures de pages multiples ayant demandé le lien zHyper.
QJSTHLCICNTW	Entier non signé	4	Nombre de pages écrites où l'écriture a demandé le lien zHyper.
QJSTHLSCIS	Entier non signé	4	Nombre d'écritures de page qui ont correctement utilisé le lien zHyper. C'est-à-dire que l'écriture a été effectuée de manière synchrone.
QJSTHLMCIS	Entier non signé	4	Nombre d'écritures de pages multiples qui ont correctement utilisé le lien zHyper. C'est-à-dire que l'écriture a été effectuée de manière synchrone.
QJSTHLCICNTS	Entier non signé	4	Nombre de pages écrites avec succès à l'aide du lien zHyper. C'est-à-dire que l'écriture a été effectuée de manière synchrone.

Nom	Tapez	Octets de longueur	Description
QJSTHLSCIF	Entier non signé	4	Nombre d'écritures de page unique qui ont tenté d'utiliser le lien zHyper, mais l'écriture n'a pas pu être effectuée de manière synchrone. A la place, l'écriture a été effectuée de manière asynchrone.
QJSTHLMCIF	Entier non signé	4	Nombre d'écritures de pages multiples qui ont tenté d'utiliser le lien zHyper, mais l'écriture n'a pas pu être effectuée de manière synchrone. A la place, l'écriture a été effectuée de manière asynchrone.
QJSTHLICNTF	Entier non signé	4	Nombre de pages écrites dans lesquelles l'écriture a tenté d'utiliser le lien zHyper, mais l'écriture n'a pas pu être effectuée de manière synchrone. A la place, l'écriture a été effectuée de manière asynchrone.
QJSTHLSCICON CNT	Entier non signé	4	Nombre d'écritures de page unique avant l'établissement d'une session d'écriture de lien zHyper. Voir la remarque «1», à la page 350.
QJSTHLMCICON CNT	Entier non signé	4	Nombre d'écritures de pages multiples avant l'établissement d'une session d'écriture de lien zHyper. Voir la remarque «1», à la page 350.
QJSTHLCONCIC NT	Entier non signé	4	Nombre de pages écrites avant l'établissement d'une connexion zHyperLink. Voir la remarque «1», à la page 350.
QJSTHLSCIWTM AX	Entier non signé	8	Temps d'écriture de lien zHyperle plus long dans l'intervalle, page unique.
QJSTHLSCIWTM IN	Entier non signé	8	Temps d'écriture de lien zHyperle plus court dans l'intervalle, page unique.
QJSTHLSCIWTT OT	Entier non signé	8	Temps total d'écriture de lien zHyperdans l'intervalle, page unique.
QJSTHLMCIWT MAX	Entier non signé	8	Temps d'écriture de lien zHyperle plus long dans l'intervalle, plusieurs pages.
QJSTHLMCIWT MIN	Entier non signé	8	Temps d'écriture de lien zHyperle plus court dans l'intervalle, plusieurs pages.
QJSTHLMCIWTT OT	Entier non signé	8	Temps total d'écriture de lien zHyperdans l'intervalle, sur plusieurs pages.
QJSTHLIOSQU	Entier non signé	16	Somme des carrés des temps d'écriture de lien zHyper, pour les écritures de page unique.
QJSTHLC	Entier non signé	4	Nombre de nouveaux journaux, utilisés dans cet intervalle SMF, qui sont compatibles avec zHyperLink.
QJSTHLE	Entier non signé	4	Nombre de nouveaux journaux, utilisés dans cet intervalle SMF, qui sont activés pour le lien zHyper.

**Remarque :**

1. Lorsque vous passez à une nouvelle copie de journal actif, il se peut qu'il y ait une période de temps avant qu'une session d'écriture de lien zHyperne soit établie avec l'unité de stockage à accès direct.

## **Message manager data records**

Use this topic as a reference for message manager data records.

The format of the message manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQMST)`.

The data gives you counts of different IBM MQ API requests.

## **Data manager data records**

Use this topic as a reference for the format of the Data Manager data records.

The format of the data manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIST)`.

The data gives you counts of different object requests.

## **Data manager page set data records**

Use this section as a reference for the format of the data manager page set data records

The format of the data manager page set statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQIS1)`.

The page set usage information helps to facilitate better management of local queues within a queue manager by recording information such as page set input-output rates and highest usage.

The data provides the same basic page set information as output by the MQSC **DISPLAY USAGE TYPE (PAGESET)** command, or the PCF Inquire Usage (**MQCMD\_INQUIRE\_USAGE**) command.

For example:

- The total pages
- The current used pages
- Unused persistent and nonpersistent pages
- Expansion method
- Number of extends
- Number of stripes

The data also provides some performance indicators, together with other performance information. For example:

- How many times deferred write, immediate write, and read page I/O requests happened during the SMF interval,
- The number of pages moved, elapsed time and number of read and write operations.
- How many pages written in checkpoints.
- Has expansion occurred during the SMF interval?
- How many times the page set became full.
- An indication of where new space is being allocated within the page set.

From the information displayed, you should be able to understand the general status of each page set, and consider whether you need to retune the system.

### **Related reference**

[“The SMF header” on page 343](#)

Use this topic as a reference for the format of the SMF header.

[“Self-defining sections” on page 343](#)

Use this topic as a reference for format of the self-defining sections of the SMF record.

[“Examples of SMF statistics records” on page 344](#)

Use this topic to understand some example SMF records.

## **Buffer manager data records**

Use this topic as a reference for the format of buffer manager data records.

The format of the buffer manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQPST)`.

**Note:** Buffer manager statistics records will only be created for buffer pools that are defined. If a buffer pool is defined but not used then no values will be set and its buffer manager statistics record will not contain any data.

For information about efficiently managing your buffer pools, see [“Managing your buffer pools” on page 353](#).

When interpreting the statistics, you are recommended to consider the following factors because the values of these fields can be used to improve the performance of your system:

1. If QPSTSOS, QPSTDMC, or QPSTIMW is greater than zero, you should either increase the size of the buffer pool or reallocate the page sets to different buffer pools.
  - QPSTSOS is the number of times that there were no buffers available for page get requests. If QPSTSOS ever becomes nonzero, it shows that IBM MQ is under severe stress. The buffer pool size should be increased. If increasing the buffer pool size does not make the value of QPSTSOS zero, there might be I/O contention on the DASD page sets.
  - QPSTDMC is the number of updates that were performed synchronously because there was either more than 95% of the pages in the buffer pool waiting for write I/O, or there was less than 5% of the buffer pool available for read requests. If this number is not zero, the buffer pool might be too small and should be enlarged. If increasing the buffer pool size does not reduce QPSTDMC to zero, there might be I/O contention on the DASD page sets.
  - QPSTIMW is a count of the number of times pages were written out synchronously. If QPSTDMC is zero, QPSTIMW is the number of times pages were found on the queue waiting for write I/O that had been there for at least two checkpoints.
2. For buffer pool zero and buffer pools that contain short-lived messages:
  - QPSTDWT should be zero, and the percentage QPSTCBSL/QPSTNBUF should be greater than 15%.  
QPSTDWT is the number of times the asynchronous write processor was started because there was either more than 85% of the pages in the buffer pool waiting for write I/O, or there was less than 15% of the buffer pool available for read requests. Increasing the buffer pool size should reduce this value. If it does not, the pattern of access is one of long delays between puts and gets.
  - QPSTTPW might be greater than zero due to checkpointing activity.
  - QPSTRIO should be zero unless messages are being read from a page set after the queue manager is restarted.

The ratio of QPSTRIO to QPSTGETP shows the efficiency of page retrieval within the buffer pool. Increasing the buffer pool size should decrease this ratio and, therefore, increase the page retrieval efficiency. If this does not happen, it indicates that pages are not being frequently reaccessed. This implies a transaction pattern where there is a long delay between messages being put and then later retrieved.

The ratio of QPSTGETN to QPSTGETP indicates the number of times an empty page, as opposed to a non-empty page, has been requested. This ratio is more an indication of transaction pattern, than a value that can be used to tune the system.

- If QPSTSTL has a value greater than zero, this indicates that pages that have not been used before are now being used. This might be caused by an increased message rate, messages not being processed as fast as they were previously (leading to a buildup of messages), or larger messages being used.

QPSTSTL is a count of the number of times a page access request did not find the page already in the buffer pool. Again, the lower the ratio of QPSTSTL to (QPSTGETP + QPSTGETN) is, the higher the



page retrieval efficiency. Increasing the buffer pool size should decrease this ratio but, if it does not, it is an indication that there are long delays between puts and gets.

- You are recommended to have sufficient buffers to handle your peak message rate.
3. For buffer pools with long-lived messages, where there are more messages than can fit into the buffer pool:
- $(QPSTRIO+QPSTWIO)/Statistics$  interval is the I/O rate to page sets. If this value is high, you should consider using multiple page sets on different volumes to allow I/O to be carried out in parallel.
  - Over the period of time that the messages are processed (for example, if messages are written to a queue during the day and processed overnight) the number of read I/Os (QPSTRIO) should be approximately the total number of pages written (QPSTPW). This shows that one page is read for every page written.

If QPSTRIO is much larger than QPSTPW, this shows that pages are being read in multiple times. This might be a result of the application using MQGET by *MsgId* or *CorrelId* when the queue is not indexed, or browsing messages on the queue using get next.

The following actions might relieve this problem:

- a. Increase the size of the buffer pool so that there are enough pages to hold the queue, in addition to any changed pages.
- b. Use the INDXTYPE queue attribute, which allows a queue to be indexed by *MsgId* or *CorrelId* and eliminates the need for a sequential scan of the queue.
- c. Change the design of the application to eliminate the use of MQGET with *MsgId* or *CorrelId*, or the get next with browse option.

**Note:** Applications using long-lived messages typically process the first available message and do not use MQGET with *MsgId* or *CorrelId*, and they might browse only the first available message.

- d. Move page sets to a different buffer pool to reduce contention between messages from different applications.

### Managing your buffer pools

To manage your buffer pools efficiently, you must consider the factors that affect the buffer pool I/O operations and also the statistics associated with the buffer pools.

The following factors affect buffer pool I/O operations.

- If a page containing the required data is not found in the buffer pool, it is read in synchronously to an available buffer from its DASD page set.
- Whenever a page is updated, it is put on an internal queue of pages to be (potentially) written out to DASD. This means that the buffer used by that page is unavailable for use by any other page until the buffer has been written to DASD.
- If the number of pages queued to be written to DASD exceeds 85% of the total number of buffers in the pool, an asynchronous write processor is started to put the buffers to DASD.

Similarly, if the number of buffers available for page get requests become less than 15% of the total number of buffers in the pool, the asynchronous write processor is started to perform the write I/O operations.

The write processor stops when the number of pages queued to be written to DASD has fallen to 75% of the total number of buffers in the pool.

- If the number of pages queued for writing to DASD exceeds 95% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

Similarly, if the number of buffers available for page get requests becomes less than 5% of the total number of buffers in the pool, all updates result in a synchronous write of the page to DASD.

- If the number of buffers available for page get requests ever reaches zero, a transaction that encounters this condition is suspended until the asynchronous write processor has finished.

- If a page is frequently updated, the page spends most of its time on the queue of pages waiting to be written to DASD. Because this queue is in least recently used order, it is possible that a frequently updated page placed on this least recently used queue is never written out to DASD. For this reason, at the time of update, if the page is found to have been waiting on the write operation to DASD queue for at least two checkpoints, it is synchronously written to DASD. Updating occurs at checkpoint time and is suspended until the asynchronous write processor has finished.

The aim of this algorithm is to maximize the time pages spend in buffer pool memory while allowing the system to function if the system load puts the buffer pool usage under stress.

### **Lock manager data records**

Use this topic as a reference to the format of the lock manager data records.

The format of the lock manager statistics record is described in assembler macro `thlqual.SCSQMACS(CSQDQLST)`.

The records contain data about the following information:

- The number of lock get requests and lock release requests.
- The number of times a lock get request determined that the requested lock was already held.

### **Db2 manager data records**

Use this topic as a reference to the format of the Db2 manager data records.

The format of the Db2 manager statistics record is described in the following table and in assembler macro `thlqual.SCSQMACS(CSQDQ5ST)` and C header file `thlqual.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `q5st`, `q5stid`.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	668	Q5ST	Db2 manager statistics
0	0	Bitstring	2	Q5STID	Control block identifier
2	2	Integer	2	Q5STLL	Control block length
4	4	Character	4	Q5STEYEC	Control block eye catcher
8	8	Character	660	Q5STZERO	QMST part cleared on occasion
8	8	Integer	4	NUMTASK	Number of server tasks
12	C	Integer	4	ACTTASK	Number of active server tasks
16	10	Integer	4	CONNCNT	Number of connect requests
20	14	Integer	4	DISCCNT	Number of disconnect requests
24	18	Integer	4	DHIGMAX	Max. request queue depth
28	1C	Integer	4	ABNDCNT	Number of Db2SRV task abends
32	20	Integer	4	REQUCNT	Number of requests queued
36	24	Integer	4	DEADCNT	Number of deadlock timeouts
40	28	Integer	4	DELECNT	Number of delete requests
44	2C	Integer	4	LISTCNT	Number of list requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
48	30	Integer	4	READCNT	Number of read requests
52	34	Integer	4	UPDTCNT	Number of update requests
56	38	Integer	4	WRITCNT	Number of write requests
60	3C	Integer	4	SCSSEL	SCST (shared-channel-status) selects
64	40	Integer	4	SCSINS	SCST inserts
68	44	Integer	4	SCSUPD	SCST updates
72	48	Integer	4	SCSDEL	SCST deletes
76	4C	Integer	4	SSKSEL	SSKT (shared-sync-key) selects
80	50	Integer	4	SSKINS	SSKT inserts
84	54	Integer	4	SSKDEL	SSKT deletes
88	58	Integer	4	SCSBFTS	SCST number of times buffer too small
92	5C	Integer	4	SCSMAXR	SCST maximum rows on query
96	60	Integer	4	* (2)	Reserved
104	68	Character	8	DELETCUW	Cumulative STCK difference - Thread delete
112	70	Character	8	DELETMXW	Maximum STCK difference - Thread delete
120	78	Character	8	DELESCUW	Cumulative STCK difference - SQL delete
128	80	Character	8	DELESMXW	Maximum STCK difference - SQL delete
136	88	Character	8	LISTTCUW	Cumulative STCK difference - Thread list
144	90	Character	8	LISTTMXW	Maximum STCK difference - Thread list
152	98	Character	8	LISTSCUW	Cumulative STCK difference - SQL list
160	A0	Character	8	LISTSMXW	Maximum STCK difference - SQL list
168	A8	Character	8	READTCUW	Cumulative STCK difference - Thread read
176	B0	Character	8	READTMXW	Maximum STCK difference - Thread read
184	B8	Character	8	READSCUW	Cumulative STCK difference - SQL read
192	C0	Character	8	READSMXW	Maximum STCK difference - SQL read
200	C8	Character	8	UPDTTCUW	Cumulative STCK difference - Thread update
208	D0	Character	8	UPDTTMXW	Maximum STCK difference - Thread update
216	D8	Character	8	UPDTSCUW	Cumulative STCK difference - SQL update
224	E0	Character	8	UPDTSMXW	Maximum STCK difference - SQL update
232	E8	Character	8	WRITTCUW	Cumulative STCK difference - Thread write
240	F0	Character	8	WRITTMXW	Maximum STCK difference - Thread write
248	F8	Character	8	WRITSCUW	Cumulative STCK difference - SQL write
256	100	Character	8	WRITSMXW	Maximum STCK difference - SQL write
264	108	Character	8	SCSSTCUW	Cumulative STCK difference - Thread select

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
272	110	Character	8	SCSSTMXW	Maximum STCK difference - Thread select
280	118	Character	8	SCSSSCUW	Cumulative STCK difference - SQL select
288	120	Character	8	SCSSSMXW	Maximum STCK difference - SQL select
296	128	Character	8	SCSITCUW	Cumulative STCK difference - Thread insert
304	130	Character	8	SCSITMXW	Maximum STCK difference - Thread insert
312	138	Character	8	SCSISCUW	Cumulative STCK difference - SQL insert
320	140	Character	8	SCSISMXW	Maximum STCK difference - SQL insert
328	148	Character	8	SCSUTCUW	Cumulative STCK difference - Thread update
336	150	Character	8	SCSUTMXW	Maximum STCK difference - Thread update
344	158	Character	8	SCSUSCUW	Cumulative STCK difference - SQL update
352	160	Character	8	SCSUSMXW	Maximum STCK difference - SQL update
360	168	Character	8	SCSDTCUW	Cumulative STCK difference - Thread delete
368	170	Character	8	SCSDTMXW	Maximum STCK difference - Thread delete
376	178	Character	8	SCSDSCUW	Cumulative STCK difference - SQL delete
384	180	Character	8	SCSDSMXW	Maximum STCK difference - SQL delete
392	188	Character	8	SSKSTCUW	Cumulative STCK difference - Thread select
400	190	Character	8	SSKSTMXW	Maximum STCK difference - Thread select
408	198	Character	8	SSKSSCUW	Cumulative STCK difference - SQL select
416	1A0	Character	8	SSKSSMXW	Maximum STCK difference - SQL select
424	1A8	Character	8	SSKITCUW	Cumulative STCK difference - Thread insert
432	1B0	Character	8	SSKITMXW	Maximum STCK difference - Thread insert
440	1B8	Character	8	SSKISCUW	Cumulative STCK difference - SQL insert
448	1C0	Character	8	SSKISMXW	Maximum STCK difference - SQL insert
456	1C8	Character	8	SSKDTCUW	Cumulative STCK difference - Thread delete
464	1D0	Character	8	SSKDTCUW	Maximum STCK difference - Thread delete
472	1D8	Character	8	SSKDSCUW	Cumulative STCK difference - SQL delete
480	1E0	Character	8	SSKDSCUW	Maximum STCK difference - SQL delete
488	1E8	Integer	4	LMSSEL	Number of Db2 BLOB read requests
492	1EC	Integer	4	LMSINS	Number of Db2 BLOB insert requests
496	1F0	Integer	4	LMSUPD	Number of Db2 BLOB update requests
500	1F4	Integer	4	LMSDEL	Number of Db2 BLOB delete requests
504	1F8	Integer	4	LMSLIS	Number of Db2 BLOB list requests
508	1FC	64 bit integer	8	LMSSTCUW	Total elapsed time for all thread read BLOB requests

Table 36. Db2 statistics record (Q5ST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
516	204	64 bit integer	8	LMSSTMXW	Maximum elapsed time for a thread read BLOB request
524	20C	64 bit integer	8	LMSSSCUW	Total elapsed time for all SQL read BLOB requests
532	214	64 bit integer	8	LMSSSMXW	Maximum elapsed time for an SQL read BLOB request
540	21C	64 bit integer	8	LMSITCUW	Total elapsed time for all thread insert BLOB requests
548	224	64 bit integer	8	LMSITMXW	Maximum elapsed time for a thread insert BLOB request
556	22C	64 bit integer	8	LMSISCUW	Total elapsed time for all SQL insert BLOB requests
564	234	64 bit integer	8	LMSISMXW	Maximum elapsed time for an SQL insert BLOB request
572	23C	64 bit integer	8	LMSUTCW	Total elapsed time for all thread update BLOB requests
580	244	64 bit integer	8	LMSUTMXW	Maximum elapsed time for a thread update BLOB request
588	24C	64 bit integer	8	LMSUSCUW	Total elapsed time for all SQL update BLOB requests
596	254	64 bit integer	8	LMSUSMXW	Maximum elapsed time for an SQL update BLOB request
604	25C	64 bit integer	8	LMSDTCW	Total elapsed time for all thread delete BLOB requests
612	264	64 bit integer	8	LMSDTMXW	Maximum elapsed time for a thread delete BLOB request
620	26C	64 bit integer	8	LMSDSCW	Total elapsed time for all SQL delete BLOB requests
628	274	64 bit integer	8	LMSDSMXW	Maximum elapsed time for an SQL delete BLOB request
636	27C	64 bit integer	8	LMSLTCW	Total elapsed time for all thread list BLOB requests
644	284	64 bit integer	8	LMSLTMXW	Maximum elapsed time for a thread list BLOB request
652	28C	64 bit integer	8	LMSLSCW	Total elapsed time for all SQL list BLOB requests
660	294	64 bit integer	8	LMSLSMXW	Maximum elapsed time for an SQL list BLOB request

The data contains counts for each request type that the Db2 resource manager supports. For these request types, maximum and cumulative elapse times are kept for the following:

- The time spent in the Db2 resource manager as a whole (called the thread time).

- The time that was spent performing the RRSAF and SQL parts of the request (a subset of the thread time called the SQL time).

Information is also provided for:

- The number of server tasks attached.
- The maximum overall request depth against any of the server tasks.
- The number of times any of the server task requests terminated abnormally.

If the abnormal termination count is not zero, a requeue count is provided indicating the number of queued requests that were requeued to other server tasks as a result of the abnormal termination.

If the average thread time is significantly greater than the average SQL time, this might indicate that thread requests are spending an excessive amount of time waiting for a server task to process the SQL part of the request. If this is the case, examine the DHIGMAX field and, if the value is greater than one, consider increasing the number of Db2 server tasks specified in the QSGDATA parameter of the CSQ6SYSP system parameter macro.

### **Coupling facility manager data records**

Use this topic as a reference to the format of the coupling facility manager data records.

The format of the coupling facility manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQEST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qest, qestid.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	4104	QEST	CF manager statistics
0	0	Bitstring	2	QESTID	Control block identifier
2	2	Integer	2	QESTLL	Control block length
4	4	Character	4	QESTEYEC	Control block eye catcher
8	8	Character	4096	QESTZERO	QEST part cleared on occasion
8	8	Character	64	QESTSTUC (0:63)	Array (one entry per structure)
8	8	Character	12	QESTSTR	Structure name
20	14	Integer	4	QESTSTRN	Structure number
24	18	Integer	4	QESTCSEC	Number of IXLLSTE calls
28	1C	Integer	4	QESTCMEC	Number of IXLLSTM calls
32	20	Character	8	QESTSSTC	Time spent doing IXLLSTE calls
40	28	Character	8	QESTMSTC	Time spent doing IXLLSTM calls
48	30	Integer	4	QESTRSEC	Number of IXLLSTE redrives
52	34	Integer	4	QESTRMEC	Number of IXLLSTM redrives
56	38	Integer	4	QESTSFUL	Number of structure fulls

Table 37. Coupling facility statistics record (QEST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
60	3C	Integer	4	QESTMNUS	Maximum number of entries in use
64	40	Integer	4	QESTMLUS	Maximum number of elements in use
68	44	Character	4	*	Reserved
4104	1008	Character	0	*	End of control block

The data contains information for each coupling facility list structure, including the CSQ\_ADMIN structure, that the queue manager could connect to during the statistics interval. The information for each structure includes the following:

- The number of and cumulative elapsed times for IXLLSTE and IXLLSTM requests.
- The number of times a request had to be retried because of a timeout.
- The number of times a 'structure full' condition occurred.

### Topic manager data records

Use this topic as a reference to the format of the topic manager data records.

The format of the topic manager statistics record is described in the following table and in assembler macro thlqual.SCSQMACS(CSQDQTST) and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qtst, qtstid.

Table 38. Topic manager statistics record (QTST)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
0	0	Structure	96	QTST	Topic manager statistics
0	0	Bitstring	2	QTSTID	Control block identifier
2	2	Integer	2	QTSTLL	Control block length
4	4	Character	4	TESTEYEC	Control block eye catcher
8	8	Character	88	QTSTZERO	QTST part cleared on occasion
8	8	Integer	4	QTSTSTOT	Total subscription requests
12	0C	Integer	4	QTSTSDUR	Durable subscription requests
16	10	Integer	4	QTSTSHIG (1:3)	Subscription high water mark array (API, ADMIN, PROXY)
28	1C	Integer	4	QTSTSLOW (1:3)	Subscription low water mark array (API, ADMIN, PROXY)
40	28	Integer	4	QTSTSEXP	Subscriptions expired
44	2C	Integer	4	QTSTTMSG	Total messages put to Sub queue
48	30	Integer	4	QTSTSPHW	Single publish subscriber high water mark
52	34	Integer	4	QTSTPTOT (1:3)	Total Publication requests (API, ADMIN, PROXY)
64	40	Integer	4	QTSTPTHI	Total publish high water mark

Table 38. Topic manager statistics record (QTST) (continued)

Offset: Dec	Offset: Hex	Type	Len	Name	Description
68	44	Integer	4	QTSTPTLO	Total publish low water mark
72	48	Integer	4	QTSTPNOS	Count of publishes to no subscriber
76	4C	Integer	4	*	Reserved
80	50	Bitstring	8	QTSTETHW	Elapse time HW on publish
88	58	Bitstring	8	QTSTETTO	Elapse time total on publish

### Coupling facility manager SMDS data records

Use this topic as a reference to the format of the coupling facility manager shared message data set (SMDS) data records.

The format of the coupling facility manager shared message data set (SMDS) statistics record is described in assembler macro `thlqual.SCSQMACS (CSQDQESD)` and in C header file `thlqual.SCSQC370 (CSQDSMFC)`.

The statistics provide information about the utilization of the owned shared message data set, I/O activity for the group of shared message data sets, and SMDS buffer utilization.

If the queue manager was not started as a member of a queue sharing group, no data is recorded in this record.

### Layout of channel initiator SMF type 115 records

The layout of channel initiator statistics data (SMF type 115, subtype 231) records is described in this topic.

## Self-defining section

The self-defining section for the channel initiator statistics data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure `qwsx` in the C programming language header file `thlqual.SCSQC370 (CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS (CSQDQWSX)`.

Table 39 on page 360 shows the format of the self-defining section.

Table 39. Structure of the channel initiator statistics self-defining section

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Integer	4	QWSX0PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWSX0PSL	Length of the QWHS
6	6	Integer	2	QWSX0PSN	Number of instances of QWHS
8	8	Integer	4	QWSX0R1O	Offset from the start of the SMF record to the first channel initiator control information block (QCCT)
12	C	Integer	2	QWSX0R1L	Length of the QCCT
14	E	Integer	2	QWSX0R1N	Number of instances of QCCT



Table 39. Structure of the channel initiator statistics self-defining section (continued)

Offset: Dec	Offset : Hex	Type	Length	Name	Description
16	10	Integer	4	QWSX0R2O	Offset from the start of the SMF record to the first dispatcher task block (QCT_DSP)
20	14	Integer	2	QWSX0R2L	Length of the QCT_DSP
22	16	Integer	2	QWSX0R2N	Number of instances of QCT_DSP
24	18	Integer	4	QWSX0R3O	Offset from the start of the SMF record to the first adapter task block (QCT_ADP)
28	1C	Integer	2	QWSX0R3L	Length of the QCT_ADP
30	1E	Integer	2	QWSX0R3N	Number of instances of QCT_ADP
32	20	Integer	4	QWSX0R4O	Offset from the start of the SMF record to the first SSL task block (QCT_SSL)
36	24	Integer	2	QWSX0R4L	Length of the QCT_SSL
38	26	Integer	2	QWSX0R4N	Number of instances of QCT_SSL
40	28	Integer	4	QWSX0R5O	Offset from the start of the SMF record to the first DNS task block (QCT_DNS)
44	2C	Integer	2	QWSX0R5L	Length of the QCT_DNS
46	2E	Integer	2	QWSX0R5N	Number of instances of QCT_DNS

Typically one record contains all the data. If there are a large number of dispatchers, adapters, or SSL tasks, the data is split over more than one record.

If this happens, the count of instances of some type of tasks can be zero, and information about a group of tasks can be spread across multiple records. The channel initiator control information block (QCCT) is only present in the first record. For example the data could be split between two SMF records like this:

Table 40. Example data

Count	First record	Last record
QWHS	1	1
QCCT	1	0
QCT_DSP	50	5
QCT_ADP	0	10
QCT_SSL	0	3
QCT_DNS	0	1

This example shows that there were 55 dispatcher TCBs running during the SMF interval.

### Instrumentation standard header (QWHS)

The format of the QWHS is described in structure qwhs in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 115 records:

Table 41. Key fields in the QWHS

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHSMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record. The subsystem ID in QWHSSSID, and the SMF interval start time in QWHSTIME, can be used to group multiple records for the same interval.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHS DURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

### Channel initiator statistics data records

Use this topic as a reference for channel initiator statistics data records.

The format of the channel initiator statistics data record contains two parts:

- The first part is the channel initiator control information block, described in assembler macro `th1qua1.SCSQMACS(CSQDQCCT)`. For further information, see [“Channel initiator control information block”](#) on page 363.
- The second part is the channel initiator task block, described in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

The channel initiator task block contains information about the four types of task within the CHINIT. For further information, see:

- [“Dispatcher tasks”](#) on page 364
- [“Adapter tasks”](#) on page 365
- [“Domain Name Server \(DNS\) task”](#) on page 366
- [“SSL tasks”](#) on page 367

Each task includes:

- The elapsed time that the task spent processing requests in the interval (*qcteltm*)
- The CPU time used by the task in the interval, which is made up of CPU used while processing requests and CPU used between requests (*qctcptm*)
- The total wait time of this task in the interval (*qctwttm*)
- The number of requests in the interval (*qctreqn*)

You can use this information to see how busy the task was, and determine whether you need to add more tasks based on the analysis.

For TLS and DNS tasks, the duration of the longest request (*qctlgdu*, *qctlsdu*) and the time of day when this occurred (*qctlgdm*, *qctlsdm*) are also included.

These can be useful to identify when channel requests took a long time. For example, a DNS lookup request going to a server outside of your enterprise taking seconds rather than milliseconds.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qctelmt*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

The example accounting data in the following tasks has been formatted using IBM MQ SupportPac MP1B.

Both of the parts are also described in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, *qcct*, *qct\_adp*.

## z/OS Channel initiator control information block

Use this topic as a reference for the channel initiator control information block.

The channel initiator control information block contains basic information for this CHINIT, including:

- CHINIT job name (*qcctjobn*)
- QSG name if it is in a queue sharing group (*qcctqsgn*)
- Peak number used of current channels (*qcctnocc*)
- Peak number used of active channels (*qcctnoac*)
- MAXCHL - maximum permitted current channels (*qcctmxcc*)
- ACTCHL - maximum permitted active channels (*qcctmxac*)
- TCPCHL - maximum permitted TCP/IP channels (*qcctmxtp*)
- LU62CHL - maximum permitted LU62 channels (*qcctmxlu*)
- **V 9.4.0** 31-bit storage used by CHINIT in the extended private region (*qcctstus*). This information is also provided by the `CSQX004I` message in the CHINIT job log.
- **V 9.4.0** 64-bit storage limit available to the CHINIT (*qcctslim*)
- **V 9.4.0** 64-bit storage used by CHINIT (*qcctstab*). This information is also provided by the `CSQX004I` message in the CHINIT job log.

The format of the channel initiator control information block is described in structure `qcct` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCCS)`.

You can use this information to see if the number of active channels is approaching the configured maximum value. Note that the number of current and active channels are the values when the record was created. So, between the two intervals there might have been more than this number of channels active.

## Channel information from SMF data

Here is an example of channel information from SMF data:

```
V 9.4.0
MV4A,MQ27,2023/10/02,11:53:02,VRM:934,
From 2023/10/02,11:52:52 to 2023/10/02,11:53:02, duration 10 seconds.
Peak number used of current channels..... 1
Peak number used of active channels ..... 1
MAXCHL. Max allowed current channels..... 9999
ACTCHL. Max allowed active channels..... 9999
TCPCHL. Max allowed TCP/IP channels..... 9999
LU62CHL. Max allowed LU62 channels..... 200
31-bit storage used..... 436 MB
64-bit storage limit.....16384 PB
64-bit storage used..... 187 MB
64-bit storage free.....16384 PB
```

You can monitor the storage usage and see whether the value is trending upwards. If the total used is approaching the total storage available, you might be running out of storage, and so might not be able to support many more channels.

If the numbers of active current channels are tending towards the maximum number of channels, you might need to increase the maximum number of channels.

## Dispatcher tasks

This topic contains example data for the dispatcher tasks statistics, and information about how to interpret the data.

The format of the dispatcher task block is described in structure `qct_dsp` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

### Example data

Task	Type	Requests	Busy %	CPU used, Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	DISP	26587	0.4	0.592463	0.1	22	127
1	DISP	26963	0.3	0.588092	0.1	22	112
2	DISP	864329	2.7	2.545668	0.3	3	28
3	DISP	26875	0.4	0.590825	0.1	22	120
4	DISP	26874	0.4	0.603285	0.1	22	123
Summ	DISP	971628	0.8	4.920332	0.1	5	38

The example data shows that there were five dispatchers. A channel is associated with a dispatcher when it starts. The channel initiator tries to distribute work across all the dispatchers when allocating a channel to a dispatcher. This example shows that one dispatcher is processing more requests than other dispatchers. This is normal, as some channels might stop, so the dispatcher is processing fewer channels, and some channels can be busier than others.

- 4.9 seconds of CPU were used by the dispatchers.
- The average request used 5 microseconds of CPU and took 38 microseconds elapsed time.
- A dispatcher is used to send and receive data over a communications network, and this is not usually dependent on external events. The average elapsed time should, therefore, be close to the average CPU time used. The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests.

The elapsed time (*qcteltm*) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time. If the CHINIT is delayed due to lack of CPU, then the ratio of average elapsed time to average CPU time is much larger, compared to when the CHINIT is not delayed for CPU.

- The average CPU used per request depends on the message traffic. For example, bigger messages use more CPU than smaller messages.

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests : `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn`
- Average ET: `qcteltm` and `qctreqn`

Usually, the number of dispatchers should be less than, or equal to, the number of processors in the LPAR. If you have more dispatchers than processors in the LPAR they might compete for CPU resources. For more information about tuning your system, see [SupportPac MP16](#).

Channels have an affinity to a dispatcher, so you might find that some dispatchers process many more requests than another dispatcher.

You can use the ALTER QMGR CHIDISPS() command to change the number of dispatchers used. Any change comes into effect the next time the channel initiator is started.

## Adapter tasks

This topic contains example data for the adapter tasks statistics, and information about how to interpret the data.

The format of the adapter task block is described in structure `qct_adp` in the C programming language header file `th1qua1.SCSQC370(CSQDSMFC)`, and in assembler macro `th1qua1.SCSQMACS(CSQDQCTA)`.

### Example data

Task	Type	Requests	Busy %	CPU used Seconds	CPU %	"avg CPU", uSeconds	"avg ET" uSeconds
0	ADAP	470297	10.2	41.290670	4.6	88	194
1	ADAP	13907	0.6	1.589428	0.2	114	365
2	ADAP	2517	0.2	0.185325	0.0	74	746
3	ADAP	1095	0.1	0.085774	0.0	78	907
4	ADAP	535	0.1	0.040743	0.0	76	947
5	ADAP	220	0.0	0.016228	0.0	74	1175
6	ADAP	82	0.0	0.005521	0.0	67	1786
7	ADAP	80	0.0	0.004248	0.0	53	1160
Summ	ADAP	488733	1.4	43.217938	0.6	88	205

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests: `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn` average
- ET: `qcteltm` and `qctreqn`

This example shows that there were eight adapter tasks.

#### Adapter number 0

- Processed the majority of the requests (470297 out of 488733)
- Was busy 10.2% of the interval
- Used 41.3 seconds of CPU

#### Overall

The average CPU per request was 88 microseconds of CPU and took 205 microseconds

The adapters process IBM MQ requests. Some of these requests might wait, for example, for log I/O during a commit, so the average Elapsed Time per request has little meaning.

The CPU time (`qctcptm`) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (`qcteltm`) value only includes time while processing requests. This means that the CPU time may be greater than the elapsed time.

When an IBM MQ request is made the first free adapter task is used.

- If there is at least one adapter that has been little used (less than 1%) busy, you have enough adapters.
- If at least one adapter was not used, you have enough adapters defined.
- If all the adapters were used, you might need to allocate more adapters.
- If all of the adapters were used, and they were all busy for most of the interval, you need to allocate more adapters.

You can use the ALTER QMGR CHIADAPS() command to change the number of adapters used. Any changes come into effect the next time the channel initiator is started.



**Attention:** If there are too many adapters acting on a small set of queues, you might get contention within the queue manager.

### Related reference

[ALTER QMGR](#)



### Domain Name Server (DNS) task

This topic contains example data for the DNS tasks statistics, and information about how to interpret the data.

The format of the DNS task block is described in structure `qct_dns` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCTA)`.

### Example data

```
Task, Type, Requests, Busy %, CPU used, CPU %, "avg CPU", "avg ET", longest,
date,           time
0, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
Summ, DNS, 14002, 0.0, 0.122578, 0.0, 9, 11, 463, 2014/03/18,
12:56:33.987671
```

The channel initiator uses a single DNS task. The example shows that the task processed 14002 requests and on average the request used 9 microseconds of CPU and took 11 microseconds of elapsed time.

The longest DNS request took 463 microseconds elapsed time, and this occurred at 12:56:33 local time.

The fields are calculated from:

- Duration: `qwhs.qwhsdurn`
- Requests : `qctreqn`
- Busy %: `qcteltm` and duration
- CPU used: `qctcptm`
- CPU %: `qctcptm` and duration
- Average CPU: `qctcptm` and `qctreqn`
- Average ET: `qcteltm` and `qctreqn`
- Longest: `qctlgdu`
- Longest at: `qctlgtm`

The DNS task can go out of your enterprise to look up the IP address associated with a name. If the average Elapsed time is significantly more than the average CPU time used, you might have some long requests.

If the value of the longest request time is unacceptable you should work with your network team to investigate why you are having long requests. It might be that you have an invalid name in your connections.

If the DNS task is busy for 25% of the duration, consider investigating the cause further.

The CPU time (`qctcptm`) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (`qcteltm`) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

**Note:** There are requests to the DNS task that are not DNS lookups, so you might have the number of requests being greater than zero - but no longest request information.

This topic contains example data for the SSL tasks statistics, and information about how to interpret the data.

The format of the SSL task block is described in structure `qct_ssl` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQCTA)`.

### Example data

Task date,	Type,	Requests,	Busy %,	CPU used,	CPU %,	"avg CPU",	"avg ET",	longest,
time				Seconds,		uSeconds,	uSeconds,	uSeconds,
0,	SSL,	3112,	1.2,	0.248538,	0.3,	80,	362,	8864,
12:46:40.237697								2014/03/18,
1,	SSL,	3070,	1.2,	0.245433,	0.3,	80,	359,	4714,
12:46:18.938022								2014/03/18,
2,	SSL,	3170,	1.2,	0.255557,	0.3,	81,	362,	7273,
12:46:35.358145								2014/03/18,
3,	SSL,	3060,	1.2,	0.246542,	0.3,	81,	365,	13164,
12:46:44.514045								2014/03/18,
4,	SSL,	3120,	1.3,	0.251927,	0.3,	81,	373,	22438,
12:46:22.134123								2014/03/18,
Summ,	SSL,	15532,	1.2,	1.247998,	0.3,	80,	364,	22438,
12:46:22.134123								2014/03/18,

This example data shows that the average request took 364 microseconds. The longest request was for SSL task 4, took 22,438 microseconds, and occurred at 12:46:22.134123 local time.

The fields are calculated from:

- Duration: *qwhs.qwhsdurn*
- Requests : *qctreqn*
- Busy %: *qcteltn* and duration
- CPU used: *qctcptm*
- CPU %: *qctcptm* and duration
- Average CPU: *qctcptm* and *qctreqn*
- Average ET: *qcteltn* and *qctreqn*
- Longest: *qctlsdu* longest at: *qctlstm*

A running channel is associated with an SSL task, in a similar way that a channel is associated with a dispatcher. The SSL tasks can use the cryptographic coprocessors available to the LPAR. So, the elapsed time can include time spent on a coprocessor. You should monitor the average elapsed time throughout the day. If this time increases significantly during peak periods you should work with your z/OS systems programmers, as your coprocessors might be over-used.

If the SSL tasks are busy for a significant proportion of the interval, increasing the number of SSL tasks might help. If the SSL tasks are waiting for external resources such as a coprocessor, increasing the number of SSL tasks has little effect.

You can use the ALTER QMGR SSLTASKS() command to change the number of SSL tasks used. Any changes come into effect the next time the channel initiator is started.

The CPU time (*qctcptm*) value includes all CPU consumed by the task, both processing requests and between processing requests. The elapsed time (*qcteltn*) value only includes time while processing requests. This means that the CPU time might be greater than the elapsed time.

### Related reference

[ALTER QMGR](#)

Use this topic as a reference for queue (SMF type 115, subtype 216) data records. The statistics are designed to make it easier for you to monitor usage and performance of your queue over time, and give an insight into what happened with your queue during the last SMF interval. This includes all the `DISPLAY QSTATUS` information and information on message flow, expiry, high and low watermarks and more.

The format of the queue statistics data record is described in assembler macro `th1qual.SCSQMACS(CSQDQST)`.

The queue statistics record contains information on the performance of selected queues and includes the following fields:

**QQSTID – Control block identifier**

The identifier for the queue statistics control block; is always `x'D80F'`.

**QQSTLL – Length of control block**

The length of a queue statistics record.

**QQSTEYEC – Control block eyecatcher**

The eyecatcher used to make identification of the control block easier; is always `'QQST'`

**QQSTQNAM – Queue name**

The name of the queue.

**QQSTFLAG**

An array of bits containing the following information about the queue:

**QQSTDISP – Queue disposition**

This bit identifies whether the queue is of private or shared disposition. If the bit is on, then it is a shared queue.

**QQSTPART – Partial record identifier**

This bit identifies whether the record is a full or partial record. If the bit is on, then it is a partial record. When this flag is set there was an issue accessing the information on the queue, for example if there is a CF structure failure.

In a shared queue partial record, the accuracy of **qqstdpth**, **qqstmage**, and **qqstuncm** cannot be guaranteed. Therefore, the fields **qqstdpth** and **qqstmage** are populated with `x'00'` and the **qqstuncm** flag is not set.

In a private queue partial record, the accuracy of **qqstmage** cannot be guaranteed, therefore the field is populated with `x'00'`.

**QQSTUNCM – Uncommitted changes pending**

This bit indicates whether there are any uncommitted changes (puts and gets) pending for the queue. If the bit is on, there are uncommitted changes.

This is checked and set at the time of the SMF data collection and provides the same result as a `DISPLAY QSTATUS` command would if run at the time the SMF record was generated.

If the queue is a shared queue and `QQSTPART` is set, this bit is always off as there might have been an issue obtaining the correct value.

**QQSTPSID – Page set ID**

The page set ID where the queue is located, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a page set assigned to it, this field is set to `-1 (x'FFFF')`.

This value is correct at the time the SMF record was generated. It is possible that the page set changed during the SMF interval, in which case, the value reflected in the next SMF record will be the new page set.

**QQSTBPID – Buffer pool ID**

The buffer pool ID used by the queue, if allocated and a private queue. If the queue is a shared queue, or a private queue that does not have a buffer pool assigned to it, this field is set to `-1 (x'FFFF')`.

This value is correct at the time the SMF record was generated. It is possible that the buffer pool changed during the SMF interval. In this case, the value reflected in the SMF record is the new buffer pool.



**QQSTQSGN – QSG name**

The Queue Sharing Group name that the queue manager is a member of, if it is a shared queue. If the queue is a private queue this field is blank.

**QQSTCFST – CF Structure name**

The coupling facility (CF) structure name the queue uses if it is a shared queue. If the queue is a private queue this field is blank.

**QQSTDPH – Current depth**

The depth of the queue at the time the SMF data was captured.

If the queue is a shared queue and QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

**QQSTOPCT – Current open for output count**

The number of handles that are currently open for output for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as OPPROCS from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

**QQSTIPCT – Current open for input count**

The number of handles that are currently open for input for the queue at the time when the SMF data was captured. For shared queues, the number returned applies only to the queue manager generating the record. The number is not the total for all the queue managers in the queue sharing group.

This is the same as IPPROCS from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

**QQSTMAGE – Oldest message age**

The age, in seconds, of the oldest message on the queue.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

If QQSTPART is set, this value is always zero as there might have been an issue obtaining the correct value.

**QQSTQTST – Short term QTIME**

The interval, in microseconds, between messages being put on the queue and then being destructively read. Value based on the last few messages processed. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the first value in QTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

**QQSTQTLT – Long term QTIME**

The interval, in microseconds, between messages being put on the queue and then being destructively read. The value is based on a larger sample of the recently processed messages. For shared queues, the values shown are for measurements collected on this queue manager only.

This is the same as the second value in QTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

**QQSTLPUT – Last put date/time**

The time, in store clock format, at which the last message was put to the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

This is the same as LPUTDATE and LPUTTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a [DISPLAY QSTATUS](#) command would, if run at the time the SMF record was generated.

### **QQSTLGET – Last get date/time**

The time, in store clock format, at which the last message was retrieved from the queue since the queue manager started. For shared queues, the value shown is for messages put by this queue manager only.

A message being browsed does not count as a message being retrieved.

This is the same as LGETDATE and LGETTIME from a [DISPLAY QSTATUS](#) command.

This is checked and set at the time of the SMF data collection and provides the same result as a DISPLAY QSTATUS command would, if run at the time the SMF record was generated.

### **QQSTDPHI – Highest depth**

The highest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPHI value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPHI is set to zero.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is higher than the current value of QQSTDPHI, then it is used as the new value of QQSTDPHI.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is higher than QQSTDPHI, if so the current queue depth is used as the new value of QQSTDPHI.

This approach means that the value of QQSTDPHI does not take into account messages put by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

### **QQSTDPL0 – Lowest depth**

The lowest depth reached by the queue during the SMF interval.

For shared queues, queue managers only have partial information about the change in depth of the queue over time. The QQSTDPL0 value is based off this partial information as follows:

- At the start of the interval the value of QQSTDPL0 is set to a special value.
- The first time during the interval the queue manager obtains the depth of the queue, QQSTDPL0 to that value.
- When an application puts a message to the queue in the interval the queue manager checks the depth of the queue, including the message just being put. If this value is lower than the current value of QQSTDPL0, then it is used as the new value of QQSTDPL0.
- When SMF data for the queue is collected, the queue manager will check if the current queue depth is lower than QQSTDPL0, if so the current queue depth is used as the new value of QQSTDPL0.

This approach means that the value of QQSTDPL0 does not take into account messages got by other queue managers in the queue sharing group, unless those messages contributed to the queue depth at the point where SMF data is collected.

### **QQSTPUTS – MQPUT count**

The number of messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

### **QQSTPUT1 – MQPUT1 count**

The number of messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

### **QQSTNPPT – Non-persistent MQPUT count**

The number of non-persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTPPT – Persistent MQPUT count**

The number of persistent messages put to the queue using MQPUT during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTNPP1 – Non-persistent MQPUT1 count**

The number of non-persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTPP1 – Persistent MQPUT1 count**

The number of persistent messages put to the queue using MQPUT1 during the SMF interval.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTPUTB – MQPUT bytes**

The number of bytes of message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTPT1B – MQPUT1 bytes**

The number of bytes of message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTNPPB – Non-persistent MQPUT bytes**

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTPPB – Persistent MQPUT bytes**

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTNP1B – Non-persistent MQPUT1 bytes**

The number of bytes of non-persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTP1B – Persistent MQPUT1 bytes**

The number of bytes of persistent message data, including any message properties, put to the queue using MQPUT1 during the SMF interval. This does not include message headers in the calculation of the size.

For shared queues, the count only includes messages put through the queue manager that generated the SMF record.

**QQSTFLPT – Failed MQPUT count**

The number of MQPUT calls targeting the queue, which failed with a completion code of MQCC\_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

**QQSTFLP1 – Failed MQPUT1 count**

The number of MQPUT1 calls targeting the queue, which failed with a completion code of MQCC\_FAILED, during the SMF interval.

For shared queues, the count only includes failed puts attempted through the queue manager that generated the SMF record.

**QQSTFPTC – Fast puts to a waiting getter count**

The number of MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval.

**Note:** *Put to a waiting getter* is a technique whereby a message might not actually be put onto a queue if there is an application already waiting to get the message. Certain conditions must be satisfied for this to occur, in particular the message must be non-persistent and the putting and getting application must be processing the message outside syncpoint control.

If these conditions are met, then the message is transferred from the putting application's buffer into the getting application's buffer without actually touching the IBM MQ queue. This removes a lot of processing involved in putting the message on the queue and therefore leads to increased throughput and lower CPU costs.

**QQSTFPTB – Fast puts to a waiting getter bytes**

The number of message and properties bytes from MQPUT and MQPUT1 calls targeting the queue, that were fast put to a waiting getter during the SMF interval. This does not include message headers in the calculation of the size.

**QQSTSTRM – Streamed message count**

The number of messages that were successfully streamed from the queue during the interval. This is always zero if STREAMQ is not set for the queue.

**QQSTMSMI – Minimum message size put**

The minimum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

**QQSTMSMA – Maximum message size put**

The maximum message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

**QQSTMSAV – Average message size put**

The average message size, in bytes, put to the queue during the SMF interval.

This includes message and properties bytes, and does not include message headers such as the MQMD.

**QQSTGETS – Destructive MQGET count**

The number of messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

**QQSTNPDG – Non-persistent destructive MQGET count**

The number of non-persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

**QQSTPDG – Persistent destructive MQGET count**

The number of persistent messages got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

**QQSTGETB – Destructive MQGET byte count**

The number of message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

**QQSTNPDB – Non-persistent destructive MQGET bytes**

The number of non-persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

**QQSTPDB – Persistent destructive MQGET bytes**

The number of persistent message and properties bytes got from the queue using destructive MQGET during the SMF interval.

For shared queues, the count only includes messages got through the queue manager that generated the SMF record.

**QQSTBRWS – Non-destructive MQGET count**

The number of messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

**QQSTNPBR – Non-persistent non-destructive MQGET count**

The number of non-persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

**QQSTPBR – Persistent non-destructive MQGET count**

The number of persistent messages browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

**QQSTBRWB – Non-destructive MQGET bytes**

The number of message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

**QQSTNPBB – Non-persistent non-destructive MQGET bytes**

The number of non-persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

**QQSTPBB – Persistent non-destructive MQGET bytes**

The number of persistent message and properties bytes browsed from the queue during the SMF interval.

For shared queues, the count only includes messages browsed through the queue manager that generated the SMF record.

**QQSTFLGT – Failed destructive MQGET count**

The number of destructive MQGET calls targeting the queue, that failed with a completion code of MQCC\_FAILED, during the SMF interval.

Also included in the count is MQCC\_WARNING when accompanied by a return code of MQRC\_TRUNCATED\_MSG\_FAILED. However, not included in this count is any MQGET with a wait that receives MQRC\_NO\_MSG\_AVAILABLE.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

**QQSTNMAG – Failed destructive MQGET with MQRC\_NO\_MSG\_AVAILABLE count**

The number of destructive MQGET calls, without wait, targeting the queue, that failed with both a completion code of MQCC\_FAILED and a return code of MQRC\_NO\_MSG\_AVAILABLE, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET, without wait, attempts through the queue manager that generated the SMF record.

**QQSTTMFB – Failed destructive MQGET with MQRC\_TRUNCATED\_MSG\_FAILED count**

The number of destructive MQGET calls targeting the queue, that failed with both a completion code of MQCC\_WARNING and a return code of MQRC\_TRUNCATED\_MSG\_FAILED, during the SMF interval.

This value is a subset of QQSTFLGT.

For shared queues, the count only includes failed MQGET attempts through the queue manager that generated the SMF record.

**QQSTFLGW – No message available for destructive MQGET with a wait count**

The number of times that there is no message available for destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

**QQSTRDGW – Re-driven destructive MQGET with a wait count**

The number of times that destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to get the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to get the message, QQSTFLGW is incremented by one and the MQGET goes back into a waiting state.

**QQSTFLBW – No message available for non-destructive MQGET with a wait count**

The number of times that there is no message available for non-destructive MQGET calls with a wait, targeting the queue, causing the MQGET to continue waiting, during the SMF interval.

**QQSTRDBW – Re-driven non-destructive MQGET with a wait count**

The number of times that non-destructive MQGET calls with a wait are re-driven to check if there is a message on the queue that matches their criteria, during the SMF interval.

When a new message arrives on the queue, all eligible waiting MQGET calls are woken up to attempt to browse the message. Each MQGET with a wait that is woken up to check for a valid message increments this count by one. If any of these MQGET with wait fails to browse the message, QQSTFLBW is incremented by one and the MQGET goes back into a waiting state.

**QQSTSAGT – Destructive MQGET with MQRC\_SIGNAL\_REQUEST\_ACCEPTED count**

The number of destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC\_WARNING and a return code of MQRC\_SIGNAL\_REQUEST\_ACCEPTED, during the SMF interval.

**QQSTSABR – Non-destructive MQGET with MQRC\_SIGNAL\_REQUEST\_ACCEPTED count**

The number of non-destructive MQGET calls targeting the queue, that complete with both a completion code of MQCC\_WARNING and a return code of MQRC\_SIGNAL\_REQUEST\_ACCEPTED, during the SMF interval.

**QQSTIPHI – High watermark for IPPROC**

The highest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

**QQSTIPL0 – Low watermark for IPPROC**

The lowest number of concurrent input handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

**QQSTOPHI – High watermark for OPPROC**

The highest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

**QQSTOPLO – Low watermark for OPPROC**

The lowest number of concurrent output handles open on the queue during the SMF interval.

For shared queues, the watermark only includes handles owned through the queue manager that generated the SMF record.

**QQSTOPEN – Successful MQOPEN count**

The number of times the queue was successfully opened during the SMF interval. This does not include opens performed as part of an MQPUT1 call.

For shared queues, the count only includes when the queue is opened through the queue manager that generated the SMF record.

**QQSTCLOS – MQCLOSE count**

The number of times the queue was successfully closed using MQCLOSE, during the SMF interval.

For shared queues, the count only includes when the queue is closed through the queue manager that generated the SMF record.

**QQSTINQR – MQINQ count**

The number of MQINQ calls that completed with a completion code of MQCC\_OK or MQCC\_WARNING, during the SMF interval.

**QQSTSET – MQSET count**

The number of MQSET calls that completed with a completion code of MQCC\_OK during the SMF interval.

**QQSTEXPR – Expired messages count**

The number of expired messages cleared from the queue during the SMF interval.

This includes messages expired by an application issuing an MQGET, by the expired message scanning task or by a REFRESH QMGR TYPE(EXPIRY) command.

**QQSTRBPT – Rolled back MQPUT counts**

The number of messages that were put to the queue, that have been rolled back off the queue, during the SMF interval.

**QQSTRBGT – Rolled back MQGET counts**

The number of messages destructively read from the queue, that have been rolled back onto the queue, during the SMF interval.

 **Interpreting IBM MQ for z/OS accounting data**

IBM MQ for z/OS accounting data is written as SMF type 116 records. Use this topic as a reference to the different types of accounting data records.

IBM MQ accounting information can be collected for the following subtypes:

**0**

Message manager accounting records (how much processor time was spent processing IBM MQ API calls and the number of MQPUT and MQGET calls). This information is produced when a named task disconnects from IBM MQ, and so the information contained within the record might cover many hours.

**1**

Accounting data for each task, at thread and queue level.

**2**

Additional queue-level accounting data (if the task used more queues than could fit in the subtype 1 record).

**10**

Accounting data for channels.

**Note:** Accounting information for specific channels can be enabled or suppressed by the **STATCHL** channel attribute, and the **STATACLS** queue manager attribute.

Note that:

- Subtype 0 records are produced with accounting trace class 1.
- Subtype 1 and 2 records are produced with accounting trace class 3.
- Subtype 10 records are produced with accounting trace class 4.

### **Layout of an SMF type 116 record**

Use this topic as a reference to the format of an SMF type record.

The standard layout for SMF records involves three parts:

#### **SMF header**

Provides format, identification, and time and date information about the record itself.

#### **Self-defining section**

Defines the location and size of the individual data records within the SMF record.

#### **Data records**

The actual data from IBM MQ that you want to analyze.

For more information about SMF record formats, see [z/OS MVS System Management Facilities \(SMF\)](#).

## **The SMF header**

Table 42 on page 376 shows the format of SMF record header (SM116).

Offset : Dec	Offset: Hex	Type	Len	Name	Description	Example
0	0	Structure	28	SM116	SMF record header.	
0	0	Integer	2	SM116LEN	SMF record length.	01A4
2	2		2		Reserved.	
4	4	Integer	1	SM116FLG	System indicator.	5E
5	5	Integer	1	SM116RTY	Record type. The SMF record type, for IBM MQ accounting records this is always 116 (X'74').	74
6	6	Integer	4	SM116TME	Time when SMF moved record.	00356124
10	A	Integer	4	SM116DTE	Date when SMF moved record.	0100223F
14	E	Character	4	SM116SID	z/OS subsystem ID. Defines the z/OS subsystem on which the records were collected.	D4E5F4F1 (MV41)
18	12	Character	4	SM116SSI	IBM MQ subsystem ID.	D4D8F0F7 (MQ07)
22	16	Integer	2	SM116STF	Record subtype.	0000
24	18	Character	3	SM116REL	IBM MQ version.	F9F3F0 (930)
27	1B		1		Reserved.	
28	1C	Character	0	SM116END	End of SMF header and start of self-defining section.	

**Note:** The (hexadecimal) values in the right-hand column relate to [Figure 22 on page 378](#).



## Self-defining sections

A self-defining section of an SMF record tells you where to find an accounting record, how long it is, and how many times that type of record is repeated (with different values). The self-defining sections follow the header, at a fixed offset from the start of the SMF record.

Each self-defining section points to accounting related data. [Table 43 on page 377](#) summarizes the offsets from the start of the SMF record header.

*Table 43. Offsets to self-defining sections*

Record subtype (SMF116STF)	Source of accounting data	Offset of self-defining section		See...
		Dec	Hex	
All	Common header	28	X'1C'	<a href="#">“Common IBM MQ SMF header” on page 379</a>
0	Message manager	44	X'2C'	<a href="#">“Message manager data records” on page 381</a>
1	Thread identification record	36	X'24'	<a href="#">“Thread-level and queue-level data records” on page 382</a>
1	Thread-level accounting	44	X'2C'	<a href="#">“Thread-level and queue-level data records” on page 382</a>
1	Queue-level accounting	52	X'34'	<a href="#">“Thread-level and queue-level data records” on page 382</a> . This section is present only if the WTASWQCT field in the task-related information (WTAS) structure is non-zero.
2	Thread identification record	36	X'24'	<a href="#">“Thread-level and queue-level data records” on page 382</a>
2	Queue-level accounting	44	X'2C'	<a href="#">“Thread-level and queue-level data records” on page 382</a>
10	Channel accounting			<a href="#">“Channel accounting data records” on page 385</a>

**Note:** Other self-defining sections refer to data for IBM use only.

Each self-defining section is two fullwords long and has this format:

```
sssssssl111nnnn
```

where:

**SSSSSSSS**

Fullword containing the offset from start of the SMF record.

**LLLL**

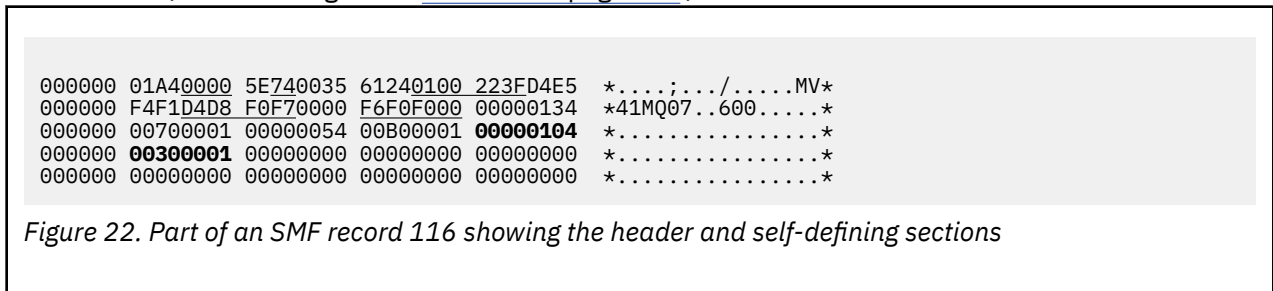
Halfword giving the length of this data record.

**NNNN**

Halfword giving the number of data records in this SMF record.

Figure 22 on page 378 shows an example of part of an SMF type 116 record. The numbers in the left-hand column represent the offset, in hexadecimal, from the start of the record. Each line corresponds to sixteen bytes of data, where each byte is two hexadecimal characters, for example 0C. The characters in the right-hand column represent the printable characters for each byte. Non-printable characters are shown by a period (.) character.

In this example, alternate fields in the SMF header are underlined to help you to see them; refer to Table 42 on page 376 to identify them. The self defining section for one of the message manager accounting data records (at the offset given in Table 43 on page 377 ) is shown in **bold**.



The self-defining section for the type of message manager accounting data is located at offset X'2C' from the start of the SMF record and contains this information:

- The offset of the message manager accounting data is located X'00000104' bytes from the start of the SMF record.
- This message manager record is X'0030' bytes long.
- There is one record (X'0001').

**Note:** Always use offsets in the self-defining sections to locate the accounting records.

**z/OS Processing type 116 SMF records**

Use this topic as a reference to the format of the processing type accounting record.

Any accounting data you collect from SMF must be processed to extract useful information. When you process the data, verify that the records are from IBM MQ and that they are the records you are expecting.

Validate the value of the following fields:

- SM116RTY, the SMF record number = X'74' (116)
- SM116STF, the record subtype, must be 0000, 0001, 0002, or 0010

Reading from the active SMF data sets (or SMF logstreams) is not supported. You must use the SMF program IFASMFDP (or IFASMF DL if logstreams are being used) to dump SMF records to a sequential data set so that they can be processed. For more information see [“Using System Management Facility” on page 335](#).

Details of the structures and fields can be found in IBM MQ SupportPac [MP1B](#).

There is a C sample program called CSQ4SMFD which prints the contents of SMF type 115 and 116 records from the sequential data set. The program is provided as source in thlqual.SCSQC37S and in executable format in thlqual.SCSQLOAD. Sample JCL is provided in thlqual.SCSQPROC(CSQ4SMFJ).

You need to update the SMFIN DD card with the name of the SMF data set. Use the z/OS command '/D SMF' to show the name of the data set, and you need to update the DUMPOUT DD card with the name for the output data set.

You also need to specify the START and END times that you require.

The following sample JCL extracts SMF records from SMF data sets and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=
//SMFIN DD DSN=xxxxxx.MANA,DISP=SHR
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
INDD(SMFIN,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

The following sample JCL extracts SMF records from the SMF log stream named in LSNAME and dumps them to the SMFOUT data set:

```
//SMFDUMP EXEC PGM=IFASMFDP,REGION=0M
//SYSPRINT DD SYSOUT=*
//SMFOUT DD DSN=xxxxxx.SMFOUT,SPACE=(CYL,(1,1)),DISP=(NEW,CATLG)
//SYSIN DD *
LSNAME(IFASMF.MQ,OPTIONS(DUMP))
OUTDD(SMFOUT,TYPE(116))
OUTDD(SMFOUT,TYPE(115))
START(1159) END(1210)
/*
```

## Common IBM MQ SMF header

Use this topic as a reference to the common IBM MQ SMF header type accounting record.

The format of this record is described in Table 44 on page 379 and in assembler macros thlqual.SCSQMACS(CSQDQWHS) and thlqual.SCSQMACS(CSQDQWHC), and C header file thlqual.SCSQC370(CSQDSMFC). The field names in C are all in lowercase, for example qwhs, qwhsnsda.

The QWHS data includes the subsystem name. For subtype 1 records, it also shows whether there are queue-level accounting records present. If the QWHSNSDA field is 3 or less, there are not, and the corresponding self-defining section (at offset X'34') is not set.

The QWHC data gives you information about the user (for example, the user ID (QWHCAID) and the type of application (QWHCATYP)). The QWHC section is completed only for subtype 0 records. The equivalent information is present in the thread identification record for subtype 1 and 2 records.

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Structure	128	QWHS	
0	0		6		Reserved
6	6	Character	1	QWHSNSDA	Number of self defining sections in the SMF records
7	7		5		Reserved
12	C	Character	4	QWHSSID	Subsystem name
16	10		24		Reserved
40	28	Character	8	QWHCAID	User ID associated with the z/OS job
48	30	Character	12	QWHCCV	Thread cross-reference
60	3C	Character	8	QWHCCN	Connection name
68	44		8		Reserved

Table 44. Structure of the common IBM MQ SMF header record QWHS (continued)

Offset: Dec	Offset : Hex	Type	Length	Name	Description
76	4C	Character	8	QWHCOPID	User ID associated with the transaction
84	54	Integer	4	QWHCATYP	Type of connecting system (1=CICS, 2=Batch or TSO, 3=IMS control region, 4=IMS MPP or BMP, 5=Command server, 6=Channel initiator, 7=RRS Batch)
88	58	Character	22	QWHCTOKN	Accounting token set to the z/OS accounting information for the user
110	6E	Character	16	QWHCNID	Network identifier
126	7E		2		Reserved

### **Combining CICS and IBM MQ performance data**

Use this topic as a reference to the combination of IBM MQ and CICS performance data.

The common IBM MQ SMF header type accounting record section, QWHCTOKN, is used to correlate CICS type 110 SMF records with IBM MQ type 116 SMF records.

CICS generates an LU6.2 unit-of-work token, for each CICS task. The token is used to generate an accounting token that is written to QWHCTOKN in the correlation header of subtype zero records.

Details are also written to the WTIDACCT section in subtype 1 and 2 records. The accounting token enables correlation between CICS and IBM MQ performance data for a transaction.

### **Thread cross-reference data**

Use this topic as a reference to the format of the thread cross-reference type accounting record.

The interpretation of the data in the thread cross-reference (QWHCCV) field varies. This depends on what the data relates to:

- CICS connections (QWHCATYP=1) - see [Table 45 on page 380](#)
- IMS connections (QWHCATYP=3 or 4) - see [Table 46 on page 380](#)
- Batch connections (QWHCATYP=2 or 7) - this field consists of binary zeros
- Others - no meaningful data

Table 45. Structure of the thread cross-reference for a CICS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	CICS thread number.
52	34	Character	4	CICS transaction name.
56	38	Integer	4	CICS task number.

Some entries contain blank characters. These apply to the task, rather than to a specific transaction.

Table 46. Structure of the thread cross-reference for an IMS system

Offset: Dec	Offset: Hex	Type	Length	Description
48	30	Character	4	IMS partition specification table (PST) region identifier.

Table 46. Structure of the thread cross-reference for an IMS system (continued)

Offset: Dec	Offset: Hex	Type	Length	Description
52	34	Character	8	IMS program specification block (PSB) name.

**z/OS Message manager data records**

Use this topic as a reference to the format of the message manager accounting records.

The message manager is the component of IBM MQ that processes all API requests. The format of the message manager accounting records is described in assembler macro `thlqual.SCSQMACS(CSQDQMAC)`.

The QMAC data gives you information about the processor time spent processing IBM MQ calls, and counts of the number of MQPUT and MQGET requests for messages of different sizes.

**Note:** A single IMS application might write two SMF records. In this case, add the figures from both records to provide the correct totals for the IMS application.

**Records containing zero processor time**

Records are sometimes produced that contain zero processor time in the QMACCPUT field. These records occur when long running tasks identified to IBM MQ either terminate or are prompted to output accounting records by accounting trace being stopped. Such tasks exist in the CICS adapter and in the channel initiator (for distributed queuing). The number of these tasks with zero processor time depends upon how much activity there has been in the system:

- For the CICS adapter, this can result in up to nine records with zero processor time.
- For the channel initiator, the number of records with zero processor time can be up to the sum of `Adapters + Dispatchers + 6`, as defined in the queue manager attributes.

These records reflect the amount of work done under the task, and can be ignored.

**z/OS Sample subtype zero accounting record**

Use this topic as a reference to the format of the subtype zero accounting records.

Figure 23 on page 381 shows a type 116, subtype zero SMF record. In this figure, the SMF record header and the QMAC accounting data record are underlined. The self-defining sections are in bold.

```

000000 01A40000 5E740035 61240100 223FD4E5 *...;.../....MV*
000010 F4F1D4D8 F0F70000 F6F0F000 00000134 *41MQ07..600....*
000020 00700001 00000054 00B00001 00000104 *.....*
000030 00300001 00000000 00000000 00000000 *.....*
000040 00000000 00000000 00000000 00000000 *.....*
000050 00000000 B478AB43 9C6C2280 B478AB47 *.....%.....*
000060 9DB47E02 00000000 04C0F631 00000001 *.=.....6.....*
000070 9880E72D 00000000 014D9540 00000000 *..X.....(. ....*
000080 08480C80 00000010 40404040 40404040 *..... *
000090 00000000 00000000 00000051 00000000 *.....*
0000A0 00000000 00000000 00000000 00000000 *.....*
0000B0 00000000 00000000 00000000 00000000 *.....*
0000C0 00000000 00000000 00000000 00000000 *.....*
0000D0 00000000 00000000 00000000 00000000 *.....*
0000E0 00000000 00000000 00000000 00000000 *.....*
0000F0 00000000 00000000 00000000 00000000 *.....*
000100 00000000 D4140030 D8D4C1C3 00000000 *...M...QMAC...*
000110 689C738D 00000050 00000000 00000050 *.....&.....&*
000120 0000000A 00000000 00000000 00000000 *.....*
000130 00000000 0024011A 00030710 02DAACF0 *......0*

```

Figure 23. Example SMF type 116, subtype zero record

## **Thread-level and queue-level data records**

Use this topic as a reference to the format of the thread-level and queue-level accounting records.

Thread level accounting records are collected for each task using IBM MQ. In addition, queue-level accounting records are gathered about each queue that the task opens. A queue-level accounting record is written for each queue that the task has used since the thread-level accounting record was last written.

If the task uses a queue that is configured with a streaming queue, there is no queue-level accounting record for the streaming queue. Instead, the accounting record for the original queue accumulates data for the data points that would have been associated with the streaming queue.

The only exception to this is that the PUTN/PUT1N value shows the number of MQPUT/MQPUT1 requests made by the application, and excludes the extra MQPUT requests made to the streaming queue.

So, for example, if an application issues a single MQPUT request, the:

- PUTN value is 1
- Elapsed time (PUTET) and CPU time (PUTCT) for the MQPUT include the time taken to put to both the primary and streaming queue
- Number of page set requests (PUTPSN) includes those for both the primary and secondary queue, and so on

For each task, data is written to SMF when the task finishes.

From IBM MQ 9.3.0 onwards, for long running tasks, data is also written at the interval specified by either the ACCTIME, or STATIME, parameter of the CSQ6SYSP system parameter macro, or by the system SMF statistics broadcast, provided that the task was running the previous time data was gathered.

Thread-level and queue-level accounting records are produced if you specify class 3 when you start the accounting trace. For example, use the following command:

```
START TRACE(ACCTG) DEST(SMF) CLASS(3)
```

The thread level accounting information is written to an SMF type 116, subtype 1 record, and is followed by queue-level records. If the task opened many queues, further queue information is written to one or more SMF type 116 subtype 2 records. A thread identification control block is included in each subtype 1 and 2 record to enable you to relate each record to the correct task. Typically, the maximum number of queue-level records in each SMF record is about 45.

The format of the thread-level accounting record is described in assembler macro `th1qua1.SCSQMACS(CSQDWTAS)`. The format of the queue-level accounting record is described in assembler macro `th1qua1.SCSQMACS(CSQDWQ)`. The format of the thread identification record is described in assembler macro `th1qua1.SCSQMACS(CSQDWTID)`. All these records are also described in C header file `th1qua1.SCSQC370(CSQDSMFC)`. The field names in C are all in lowercase, for example `wtas`, `wtasshex`.

## **Meaning of the channel names**

Use this topic as a reference to the meaning of channel names.

The channel name in the WTID is constructed as shown in the following example. In this example a sender channel exists from queue manager QM1 to queue manager QM2.

The meaning of channel names are described in the following table.

<i>Table 47. Meaning of channel names</i>		
<b>Field name</b>	<b>Meaning</b>	<b>Example</b>
For queue manager QM1 the sender channel has the following fields set:		
WTIDCCN	The job name	QM1CHIN
WTIDCHL	The channel name	QM1.QM2

Table 47. Meaning of channel names (continued)

Field name	Meaning	Example
WTIDCHLC	This is defined in the CONNAME of the channel	WINMVS2B(2162)
For queue manager QM2 the receiver channel has the following fields set:		
WTIDCCN	The job name	QM2CHIN
WTIDCHL	The channel name	QM1.QM2
WTIDCHLC	Where the channel came from	9.20.101.14

**z/OS** Sample subtype 1 and subtype 2 records

Use this topic as a reference to the format of the subtype 1 and subtype 2 accounting records.

Figure 24 on page 383 and Figure 25 on page 383 show examples of SMF type 116, subtype 1 and subtype 2 records. These two accounting records were created for a batch job that opened 80 queues. Because many queues were opened, a subtype 2 record was required to contain all the information produced.

```

000000 703C0000 5E74002D 983B0100 229FD4E5 *...;.....MV*
000010 F4F1D4D8 F0F70001 F6F0F000 00006FCC *41MQ07..600...?*
000020 00700001 0000003C 00D00001 0000010C *.....}*
000030 02C00001 000003CC 02400030 F70000D0 *.{.....7..}*
000040 E6E3C9C4 00000000 00000000 00000040 *WTID.....*
.
.
000100 00000000 00000000 7F4A4BB8 F70102C0 *....."...7..}*
000110 E6E3C1E2 B4802373 0BF07885 7F4AE718 *WTAS.....0..".X.*

```

Figure 24. Example SMF type 116, subtype 1 record

The first self-defining section starts at X'24' and is **bold** in the example; X'0000003C' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic*; X'0000010C' is the offset to the WTAS data record, X'02C0' is the length of the WTAS record, and X'0001' is the number of WTAS records.

The third self-defining section starts at X'34' and is **bold** in the example; X'000003CC' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0030' is the number of WQST records.

Figure 25 on page 383 shows an example of an SMF type 116, subtype 2 record.

```

000000 49740000 5E74002D 983B0100 229FD4E5 *...;.....MV*
000010 F4F1D4D8 F0F70002 F6F0F000 00004904 *41MQ07..600....*
000020 00700001 00000034 00D00001 00000104 *.....}*
000030 02400020 F70000D0 E6E3C9C4 00000002 *. .7..}*WTID....*
.
.
000100 7F4A4BB8 F7020240 E6D8E2E3 00000001 *"...7.. WQST....*

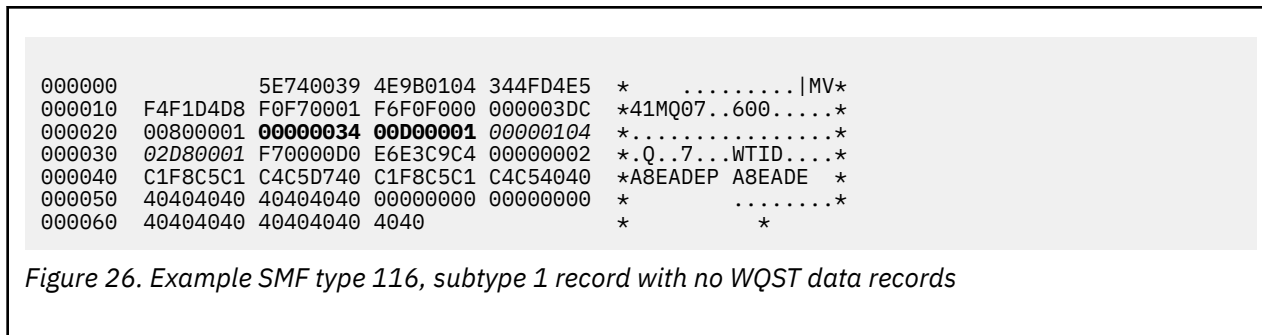
```

Figure 25. Example SMF type 116, subtype 2 record

The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic* ; X'00000104' is the offset to the first WQST data record, X'0240' is the length of the WQST record, and X'0020' is the number of WQST records.

Figure 26 on page 384 shows an example of an SMF type 116, subtype 1 record where no queues have been opened and there are consequently no self-defining sections for WQST records..



The first self-defining section starts at X'24' and is **bold** in the example; X'00000034' is the offset to the WTID data record, X'00D0' is the length of the WTID record, and X'0001' is the number of WTID records.

The second self-defining section starts at X'2C' and is in *italic* ; X'0000010C' is the offset to the WTAS data record, X'02D8' is the length of the WTAS record, and X'0001' is the number of WTAS records.

There is no self-defining section describing a WQST data record, equivalent to the third self-defining section in Figure 24 on page 383.

### **Layout of channel initiator SMF type 116 records**

The layout of channel accounting data (SMF type 116, subtype 10) records is described in this topic.

#### **Self-defining section**

The self-defining section for the channel accounting data follows the standard SMF header. It is structured in the standard triplet format. The format of the triplets is described in structure qws5 in the C programming language header file th1qua1.SCSQC370 (CSQDSMFC), and in assembler macro th1qua1.SCSQMACS (CSQDQWS5).

Table 48 on page 384 shows the format of the self-defining section.

Table 48. Structure of the channel accounting self-defining section

Offset: Dec	Offset : Hex	Type	Length	Name	Description
0	0	Integer	4	QWS50PSO	Offset from the start of the SMF record to the first instrumentation standard header (QWHS)
4	4	Integer	2	QWS50PSL	Length of the QWHS
6	6	Integer	2	QWS50PSN	Number of instances of QWHS
8	8	Integer	4	QWS50R1O	Offset from the start of the SMF record to the first channel accounting data record (QCST)
12	C	Integer	2	QWS50R1L	Length of the QCST
14	E	Integer	2	QWS50R1N	Number of instances of QCST



## Instrumentation standard header (QWHS)

The format of the QWHS is described in structure `qwhs` in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`, and in assembler macro `thlqual.SCSQMACS(CSQDQWHS)`. It contains the following key fields that are relevant to channel initiator SMF 116 records:

Name	Length	Description
QWHSNDA	1 byte	Number of self-defining sections
QWHSSSID	4 bytes	Subsystem name
QWHS SMFC	1 bit	Indicates whether there are multiple SMF records containing information for this interval. If this bit is on, information for this interval is continued in further SMF records. If this bit is off, this is the last or only record.
QWHSTIME	8 bytes	Local time of the start of the interval in STCK format
QWHS DURN	8 bytes	Duration from the start of the interval to the end of the interval in STCK format
QWHSSTCK	8 bytes	End of the interval in UTC in STCK format

### Channel accounting data records

Use this topic as a reference for channel accounting data records.

The format of the channel accounting data record is described in assembler macro `thlqual.SCSQMACS(CSQDQCST)`. The format is also described in the C programming language header file `thlqual.SCSQC370(CSQDSMFC)`. Note that the field names in C are all in lowercase, for example, `qcst`.

The channel accounting data gives you information about the status and statistics of each channel instance, including:

- Average network time (`qcstntav`)
- Average time on exit (`qcstetav`)
- Channel batch data limit (`qcstc bdl`)
- Channel batch interval (`qcstc bit`)
- Channel batch size (`qcstc bsz`)
- Channel dispatcher number (`qcstd spn`)
- Channel disposition (`qcstc hdp`)
- Channel name (`qcstc hnm`)
- Channel state (`qcstc hst`)
- Channel started time (`qcstc strt`)
- Channel status collected time (`qcstc ltm`)
- Channel stopped time (`qcstc ldt`)
- Channel type (`qcstc hty`)
- Common name (CN) from SSLCERTI (`qcstc lcn`)
- Compression rate (`qcstc pra`)
- Connection name (`qcstc nnm`)
- Current shared conversations (`qcstc scv`)

- DNS resolution time (*qcstdnrt*)
- Effective value of STATCHL parameter (*qcststcl*)
- Last message time (*qcstlmst*)
- Maximum network time (*qcstntmx*)
- Maximum time on exit (*qcstetmx*)
- Minimum network time (*qcstntmn*)
- Minimum time on exit (*qcstetmn*)
- Name of the remote queue manager or application (*qcstrqmn*)
- Number of batches (*qcstbatc*)
- Number of bytes for message data (*qcstnbyt*)
- Number of bytes for persistent message data (*qcstnpby*)
- Number of bytes received for both message data and control information (*qcstbyrc*)
- Number of bytes sent for both message data and control information (*qcstbyst*)
- Number of full batches (*qcstfuba*)
- Number of messages, or number of MQI calls (*qcstnmsg*)
- Number of persistent messages (*qcstnmsg*)
- Number of put retries (*qcstptrc*)
- Number of transmission queue becoming empty (*qcstqetc*)
- Number of transmission buffers received ( **qcstbfrc** )
- Number of transmission buffers sent (*qcstbfst*)
- Serial number from SSLPEER (*qcstslsn*)
- SSL CipherSpec (zero means TLS not used) (*qcstslcs*)
- The date and time of maximum network time (*qcstntdt*)
- The date and time of maximum time on exit (*qcstetdt*)

Note, that for the channel accounting field *qcstetmn* (Minimum time on exit) and *qcstntmn* (Minimum network time) these two fields will be initialized to the hexadecimal value of 8FFFFFFF when unused.

You can use this information to see the throughput of a channel, if the actual batches are approaching the limit, the latency of the network, information about the remote end, performance of user exit, and so on.

Here is an example of the channel accounting data which has been formatted with IBM MQ SupportPac MP1B.

The fields available are based on the display channel status command (DIS CHS) and channel statistics by IBM MQ on platforms except z/OS, with some additional fields.

The data and time of the start and end of the record in local time, and the duration

SMF interval start	2014/03/26,02:30:00
SMF interval end	2014/03/26,02:45:00
SMF interval duration	899.997759 seconds

Information about the channel

Connection name	9.20.4.159
Channel disp	PRIVATE
Channel type	RECEIVER
Channel status	CLOSING
Channel STATCHL	HIGH

Start date & time	2014/03/26,02:44:58
Channel status collect time	2014/03/26,02:45:00
Last status changed	1900/01/01,00:00:00
Last msg time	2014/03/26,02:44:59

```

Batch size                50
Messages/batch           3.3
Number of messages       1,102
Number of persistent messages 1,102
Number of batches        335
Number of full batches    0
Number of partial batches 335
Buffers sent              337
Buffers received         1,272
Message data              5,038,344  4 MB
Persistent message data  5,038,344  4 MB
Non persistent message data 0  0 B
Total bytes sent          9,852  9 KB
Total bytes received     5,043,520  4 MB
Bytes received/Batch     15,055  14 KB
Bytes sent/Batch         29  29 B
Batches/Second           1
Bytes received/message   4,576  4 KB
Bytes sent/message       8  8 B
Bytes received/second    28,019  27 KB/sec
Bytes sent/second        54  54 B/sec
Compression rate         0

```

```

The name of the queue manager at the remote end of the connection
Remote qmgr/app          MQPH
Put retry count          0

```

## Optimisation de votre réseau IBM MQ

Utilisez les conseils d'optimisation de cette section pour améliorer les performances de votre réseau de gestionnaires de files d'attente.

### Réglage des canaux de connexion client et serveur

Le paramètre par défaut pour **SHARECNV** est 10, qui autorise jusqu'à 10 conversations client pour chaque instance de canal. Toutefois, l'utilisation d'un nombre différent de conversations partagées peut être plus efficace pour les performances. Si vous n'avez pas besoin de conversations partagées ou si vous utilisez un serveur réparti, définissez **SHARECNV** sur 1. Si vous disposez d'applications client existantes qui ne s'exécutent pas correctement lorsque vous définissez **SHARECNV** sur 1 ou une valeur supérieure, définissez **SHARECNV** sur 0.

#### Pourquoi et quand exécuter cette tâche

Pour certaines configurations, l'utilisation de conversations partagées apporte des avantages significatifs. Toutefois, pour les serveurs répartis, le traitement des messages sur les canaux qui utilisent la configuration par défaut de 10 conversations partagées est en moyenne 15% plus lent que sur les canaux qui n'utilisent pas de conversations partagées. Sur une instance de canal MQI qui partage les conversations, toutes les conversations sur un socket sont reçues par la même unité d'exécution. Si les conversations partageant un socket sont toutes occupées, les unités d'exécution conversationnelles se disputent l'utilisation de l'unité d'exécution réceptrice. Le conflit entraîne des retards et, dans ce cas, il est préférable d'utiliser un plus petit nombre de conversations partagées.

Le paramètre **SHARECNV** permet d'indiquer le nombre maximal de conversations à partager sur une instance de canal client TCP/IP particulière. Pour plus de détails sur toutes les valeurs possibles, voir [Client IBM MQ pris en charge: comportement par défaut des canaux de connexion client et de connexion serveur](#).

Si vous affectez à **SHARECNV** la valeur 1 ou une valeur supérieure, vous activez les améliorations de performances suivantes:

- Pulsations bidirectionnelles
- Arrêt de la mise au repos de l'administrateur,
- Lecture anticipée
- Consommation asynchrone par les applications client

Si vous n'avez pas besoin de conversations partagées, ces deux paramètres offrent de meilleures performances:

- SHARECNV(1).
- SHARECNV(0).

#### Remarques :

- Si la valeur **SHARECNV** de connexion client ne correspond pas à la valeur **SHARECNV** de connexion serveur, la valeur la plus faible est utilisée.
- Lorsque des applications sont liées ou compilées dans une bibliothèque non réentrante, la valeur CURSHCNV(0) est négociée même si une valeur supérieure est définie dans CLNTCONN et SVRCONN.

Pour optimiser les performances d'une instance de canal donnée, effectuez l'une des étapes suivantes.

#### Procédure

- Surveillez les canaux qui utilisent la valeur SHARECNV par défaut 10.

Le paramètre par défaut de SHARECNV(10) fonctionne bien dans de nombreux scénarios, mais il peut ne pas être le paramètre optimal pour une instance de canal donnée. Par exemple, pour les serveurs répartis, le traitement des messages sur les canaux qui utilisent ce paramètre est en moyenne 15% plus lent que sur les canaux qui n'utilisent pas de conversations partagées.

Pour vous assurer que le paramètre par défaut est approprié pour une instance de canal donnée, surveillez le fonctionnement du canal avec ce paramètre.

- Définissez une valeur SHARECNV supérieure ou égale à 2 .

Vous pouvez définir SHARECNV(2) sur SHARECNV(999999999) . Pour vous assurer que le paramètre que vous choisissez est approprié pour une instance de canal donnée, surveillez le fonctionnement du canal avec le nouveau paramètre.

- Définissez la valeur SHARECNV sur 1.

Si vous n'avez pas besoin de conversations partagées, utilisez ce paramètre dans la mesure du possible. Il élimine les conflits d'utilisation de l'unité d'exécution de réception et vos applications client peuvent tirer parti des améliorations de performances décrites dans la section "A propos de cette tâche".

Avec ce paramètre, les performances du serveur réparti sont considérablement améliorées. Les améliorations des performances s'appliquent aux applications client qui émettent des appels d'attente d'obtention synchrone sans lecture anticipée ; par exemple, les appels d'attente MQGET du client C. Lorsque ces applications client sont connectées, le serveur réparti utilise moins d'unités d'exécution et moins de mémoire et le débit est augmenté.

Si un serveur est connecté à des clients qui partagent des conversations via un socket et que vous réduisez le paramètre des conversations partagées de SHARECNV(10) à SHARECNV(1), cela a les effets suivants:

- Utilisation accrue des sockets sur le serveur.
- Augmentation des instances de canal sur le serveur.

Dans ce cas, vous pouvez également choisir d'augmenter les paramètres pour **MaxChannels** et **MaxActiveChannels**.

**Remarque :** Vous pouvez également définir l'option MQCONNX , MQCNO\_NO\_CONV\_SHARING et connecter l'application à un canal avec **SHARECNV** défini sur une valeur supérieure à 1. Le résultat est le même que celui de la connexion de l'application à un canal avec **SHARECNV** défini sur 1.

- Définissez la valeur 0 pour SHARECNV .

L'instance de canal se comporte exactement comme s'il s'agissait d'un canal de connexion serveur ou client IBM WebSphere MQ 6.0 . Vous n'obtenez pas de conversations partagées, ni les améliorations de performances disponibles lorsque vous affectez à **SHARECNV** la valeur 1 ou une valeur supérieure.

Utilisez la valeur 0 uniquement si vous disposez d'applications client existantes qui ne s'exécutent pas correctement lorsque vous affectez à **SHARECNV** la valeur 1 ou une valeur supérieure.

### Concepts associés

Client IBM MQ pris en charge: comportement par défaut des canaux de connexion client et de connexion serveur

## Optimisation des réseaux de publication / abonnement distribués

Utilisez les conseils d'optimisation de cette section pour améliorer les performances de vos clusters et hiérarchies de publication / abonnement distribué IBM MQ.

### Concepts associés

«Surveillance des clusters», à la page 325

Dans un cluster, vous pouvez surveiller les messages d'application, les messages de contrôle et les journaux. Il existe des considérations de surveillance spéciales lorsque le cluster équilibre la charge entre deux ou plusieurs instances d'une file d'attente.

## Performances du cluster de publication / abonnement par routage direct

Dans les clusters de publication / abonnement routés directement, les informations telles que les rubriques de cluster et les abonnements de proxy sont envoyées à tous les membres du cluster, que tous les gestionnaires de files d'attente de cluster participent activement à la messagerie de publication / abonnement. Ce processus peut créer une charge supplémentaire importante sur le système. Pour réduire l'impact de la gestion de cluster sur les performances, vous pouvez effectuer des mises à jour aux heures creuses, définir un sous-ensemble beaucoup plus petit de gestionnaires de files d'attente impliqués dans la publication / l'abonnement et créer un cluster "chevauchant" ou passer à l'utilisation du routage via l'hôte de rubrique.

Il existe deux sources de charge de travail sur un gestionnaire de files d'attente dans un cluster de publication / abonnement:

- Gestion directe des messages pour les programmes d'application.
- Gestion des messages et des canaux nécessaires à la gestion du cluster.

Dans un cluster point-à-point typique, la charge de travail du système de cluster est largement limitée aux informations explicitement demandées par les membres du cluster, selon les besoins. Par conséquent, dans tout autre élément qu'un très grand cluster point à point, par exemple un cluster qui contient des milliers de gestionnaires de files d'attente, vous pouvez largement mettre en valeur l'effet de la gestion du cluster sur les performances. Toutefois, dans un cluster de publication / abonnement routé directement, les informations telles que les rubriques en cluster, l'appartenance à un gestionnaire de files d'attente et les abonnements de proxy sont envoyées à tous les membres du cluster, que tous les gestionnaires de files d'attente de cluster participent ou non activement à la messagerie de publication / abonnement. Cela peut créer une charge supplémentaire importante sur le système. Par conséquent, vous devez prendre en compte l'effet de la gestion de cluster sur les performances du gestionnaire de files d'attente, tant dans sa durée que dans sa taille.

## Caractéristiques de performances des clusters routés directement

Comparez un cluster point à point avec un cluster de publication / abonnement routé directement en ce qui concerne les tâches de gestion de base.

Tout d'abord, un cluster point à point:

1. Lorsqu'une nouvelle file d'attente de cluster est définie, les informations de destination sont envoyées aux gestionnaires de files d'attente de référentiel complet et envoyées uniquement aux autres membres de cluster lorsqu'ils font référence pour la première fois à une file d'attente de cluster (par exemple, lorsqu'une application tente de l'ouvrir). Ces informations sont ensuite mises en cache localement par le gestionnaire de files d'attente afin de supprimer la nécessité d'extraire à distance les informations à chaque fois que vous accédez à la file d'attente.

2. L'ajout d'un gestionnaire de files d'attente à un cluster n'affecte pas directement la charge des autres gestionnaires de files d'attente. Les informations sur le nouveau gestionnaire de files d'attente sont envoyées aux référentiels complets, mais les canaux vers le nouveau gestionnaire de files d'attente à partir d'autres gestionnaires de files d'attente du cluster sont créés et démarrés uniquement lorsque le trafic commence à circuler vers ou depuis le nouveau gestionnaire de files d'attente.

En résumé, la charge sur un gestionnaire de files d'attente dans un cluster point à point est liée au trafic de messages qu'il gère pour les programmes d'application et n'est pas directement liée à la taille du cluster.

Deuxièmement, un cluster de publication / abonnement routé directement:

1. Lorsqu'une nouvelle rubrique de cluster est définie, les informations sont envoyées aux gestionnaires de files d'attente de référentiel complet et, à partir de là, directement à tous les membres du cluster, ce qui entraîne le démarrage des canaux vers chaque membre du cluster à partir des référentiels complets s'ils ne sont pas déjà démarrés. S'il s'agit de la première rubrique en cluster direct, des informations sur tous les autres membres du gestionnaire de files d'attente du cluster sont envoyées à chaque membre du gestionnaire de files d'attente.
2. Lorsqu'un abonnement est créé à une rubrique de cluster sur une nouvelle chaîne de rubrique, les informations sont envoyées directement de ce gestionnaire de files d'attente vers tous les autres membres du cluster immédiatement, ce qui entraîne le démarrage des canaux vers chaque membre du cluster à partir de ce gestionnaire de files d'attente s'ils ne sont pas déjà démarrés.
3. Lorsqu'un nouveau gestionnaire de files d'attente rejoint un cluster existant, les informations sur toutes les rubriques en cluster (et sur tous les membres du gestionnaire de files d'attente si une rubrique de cluster directe est définie) sont envoyées au nouveau gestionnaire de files d'attente à partir des gestionnaires de files d'attente de référentiel complet. Le nouveau gestionnaire de files d'attente synchronise ensuite la connaissance de tous les abonnements aux rubriques de cluster du cluster avec tous les membres du cluster.

En résumé, la charge de gestion de cluster sur n'importe quel gestionnaire de files d'attente dans un cluster de publication / abonnement routé directement augmente avec le nombre de gestionnaires de files d'attente, de rubriques en cluster et de modifications apportées aux abonnements sur des chaînes de rubrique différentes dans le cluster, quelle que soit l'utilisation locale de ces rubriques de cluster sur chaque gestionnaire de files d'attente.

Dans un cluster de grande taille, ou dans un cluster où le taux de changement des abonnements est élevé, ce niveau de gestion de cluster peut être une surcharge importante pour tous les gestionnaires de files d'attente.

## **Réduction de l'effet de la publication / abonnement direct routé sur les performances**

Pour réduire l'impact de la gestion de cluster sur les performances d'un cluster de publication / abonnement routé directement, tenez compte des options suivantes:

- Effectuez des mises à jour de cluster, de rubrique et d'abonnement aux heures creuses de la journée.
- Définissez un sous-ensemble beaucoup plus petit de gestionnaires de files d'attente impliqués dans la publication / l'abonnement et faites en sorte qu'il s'agit d'un cluster "chevauchant". Ce cluster est ensuite le cluster dans lequel des rubriques de cluster sont définies. Bien que certains gestionnaires de files d'attente se trouvent désormais dans deux clusters, l'effet global de la publication / abonnement est réduit:
  - La taille du cluster de publication / abonnement est plus petite.
  - Les gestionnaires de files d'attente qui ne se trouvent pas dans le cluster de publication / abonnement sont beaucoup moins affectés par le trafic de gestion de cluster.

Si les options précédentes ne permettent pas de résoudre correctement vos problèmes de performances, envisagez d'utiliser un cluster de publication / abonnement *de type hôte de rubrique* routé à la place. Pour une comparaison détaillée du routage direct et du routage par hôte de rubrique dans les clusters de publication / abonnement, voir [Conception de clusters de publication / abonnement](#).

## Concepts associés

### Performances du cluster de publication / abonnement de l'hôte de rubriques

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. Ces hôtes de rubrique deviennent les gestionnaires de files d'attente de *routage* pour cette branche de l'arborescence de rubriques. De plus, les gestionnaires de files d'attente sans abonnements ou diffuseurs de publications n'ont pas besoin de se connecter aux hôtes de rubrique. Cette configuration peut réduire de manière significative le nombre de connexions entre les gestionnaires de files d'attente dans le cluster et la quantité d'informations transmises entre les gestionnaires de files d'attente.

### Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement

Un concept important dans les performances de messagerie asynchrone est *balance*. A moins que les consommateurs de messages ne soient équilibrés avec les expéditeurs de messages, il existe un risque qu'un arriéré de messages non consommés puisse s'accumuler et affecter sérieusement les performances de plusieurs applications.

### Performances des abonnements dans les réseaux de publication / abonnement

La publication / l'abonnement distribué dans IBM MQ fonctionne en propageant la connaissance de l'emplacement où des abonnements à des chaînes de rubrique différentes ont été créés dans le réseau du gestionnaire de files d'attente. Cela permet au gestionnaire de files d'attente sur lequel un message est publié d'identifier les autres gestionnaires de files d'attente qui nécessitent une copie du message publié pour correspondre à leurs abonnements.

## Performances du cluster de publication / abonnement de l'hôte de rubriques

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. Ces hôtes de rubrique deviennent les gestionnaires de files d'attente de *routage* pour cette branche de l'arborescence de rubriques. De plus, les gestionnaires de files d'attente sans abonnements ou diffuseurs de publications n'ont pas besoin de se connecter aux hôtes de rubrique. Cette configuration peut réduire de manière significative le nombre de connexions entre les gestionnaires de files d'attente dans le cluster et la quantité d'informations transmises entre les gestionnaires de files d'attente.

Un cluster de publication / abonnement routé par un hôte de rubrique se comporte comme suit:

- Les rubriques sont définies manuellement sur des gestionnaires de files d'attente *d'hôte de rubrique* individuels dans le cluster.
- Lorsqu'un abonnement est effectué sur un gestionnaire de files d'attente de cluster, les abonnements de proxy sont créés uniquement sur les hôtes de rubrique.
- Lorsqu'une application publie des informations dans une rubrique, le gestionnaire de files d'attente de réception transmet la publication à un gestionnaire de files d'attente qui héberge la rubrique. L'hôte de rubrique envoie ensuite la publication à tous les gestionnaires de files d'attente du cluster qui ont des abonnements valides à la rubrique.

Pour une présentation plus détaillée du routage par hôte de rubrique, voir [Routage par hôte de rubrique dans les clusters](#).

Pour de nombreuses configurations, le routage par hôte de rubrique est une topologie plus appropriée que le *routage direct* car il offre les avantages suivants:

- Evolutivité améliorée des clusters de grande taille. Seuls les gestionnaires de files d'attente hôte de rubrique doivent pouvoir se connecter à tous les autres gestionnaires de files d'attente dans le cluster. Par conséquent, il existe moins de canaux entre les gestionnaires de files d'attente et moins de trafic administratif de publication/abonnement entre les gestionnaires de files d'attente que pour dans le cas du routage direct. Lorsque les abonnements changent sur un gestionnaire de files d'attente, seuls les gestionnaires de files d'attente hôte de rubrique doivent en être informés.
- Contrôle accru sur la configuration physique. Avec le routage direct, tous les gestionnaires de files d'attente assument tous les rôles et doivent donc disposer de capacités égales. Avec le routage via un hôte de rubrique, vous sélectionnez 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 matériel adapté et vous pouvez utiliser des systèmes moins puissants pour les autres gestionnaires.

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.
- 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 :** Pour certaines configurations, il est utile de supprimer cette contrainte, comme décrit dans [Routage de l'hôte de rubrique à l'aide de diffuseurs de publications ou d'abonnés centralisés](#).

- 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.

## **Routage via un hôte de rubriques avec des diffuseurs de publications ou des abonnés centralisés**

Pour supprimer les "tronçons" supplémentaires générés lorsque les publications sont toujours acheminées vers des abonnements via un gestionnaire de files d'attente hôte de rubrique, configurez les diffuseurs de publications ou les abonnements sur le même gestionnaire de files d'attente qui héberge la rubrique. Cette approche offre un maximum d'avantages en termes de performances dans les deux cas suivants:

- Rubriques avec de nombreux diffuseurs de publications et peu d'abonnements. Dans ce cas, hébergez les abonnements sur le gestionnaire de files d'attente hôte de rubrique.
- Rubriques avec peu de diffuseurs de publications et de nombreux abonnements. Dans ce cas, hébergez les diffuseurs sur le gestionnaire de files d'attente hôte de rubrique.

La figure suivante présente un gestionnaire de files d'attente hôte de rubrique qui héberge également les abonnements. Cette approche supprime le "saut" supplémentaire entre le diffuseur de publications et l'abonné et réduit le partage inutile des connaissances sur les abonnements entre tous les membres du cluster:



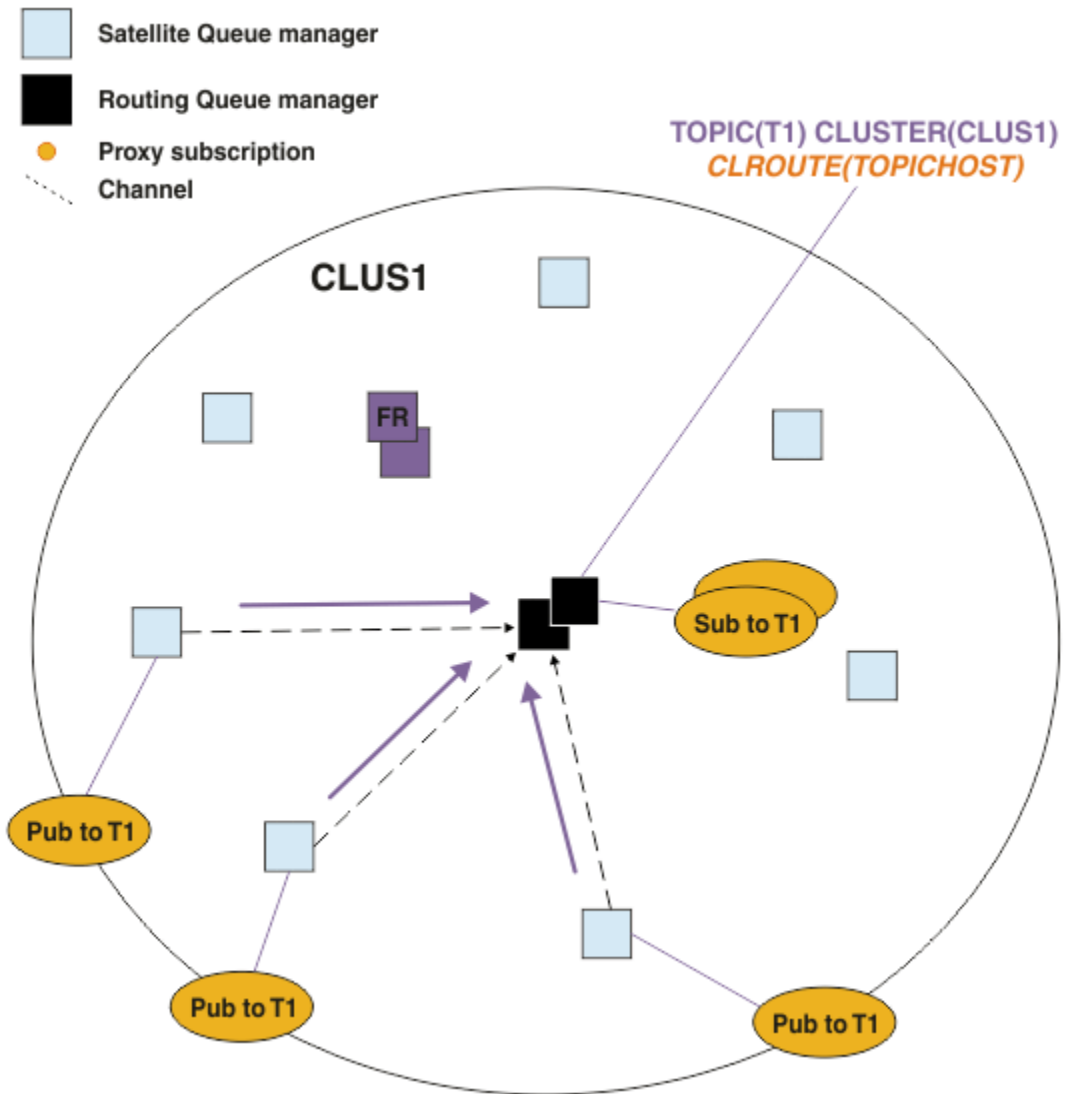


Figure 27. Hébergement d'abonnements sur un gestionnaire de files d'attente hôte de rubrique

La figure suivante présente un gestionnaire de files d'attente hôte de rubrique qui héberge également les diffuseurs de publications. Cette approche supprime le "saut" supplémentaire entre le diffuseur de publications et l'abonné et réduit le partage inutile des connaissances sur les abonnements entre tous les membres du cluster:

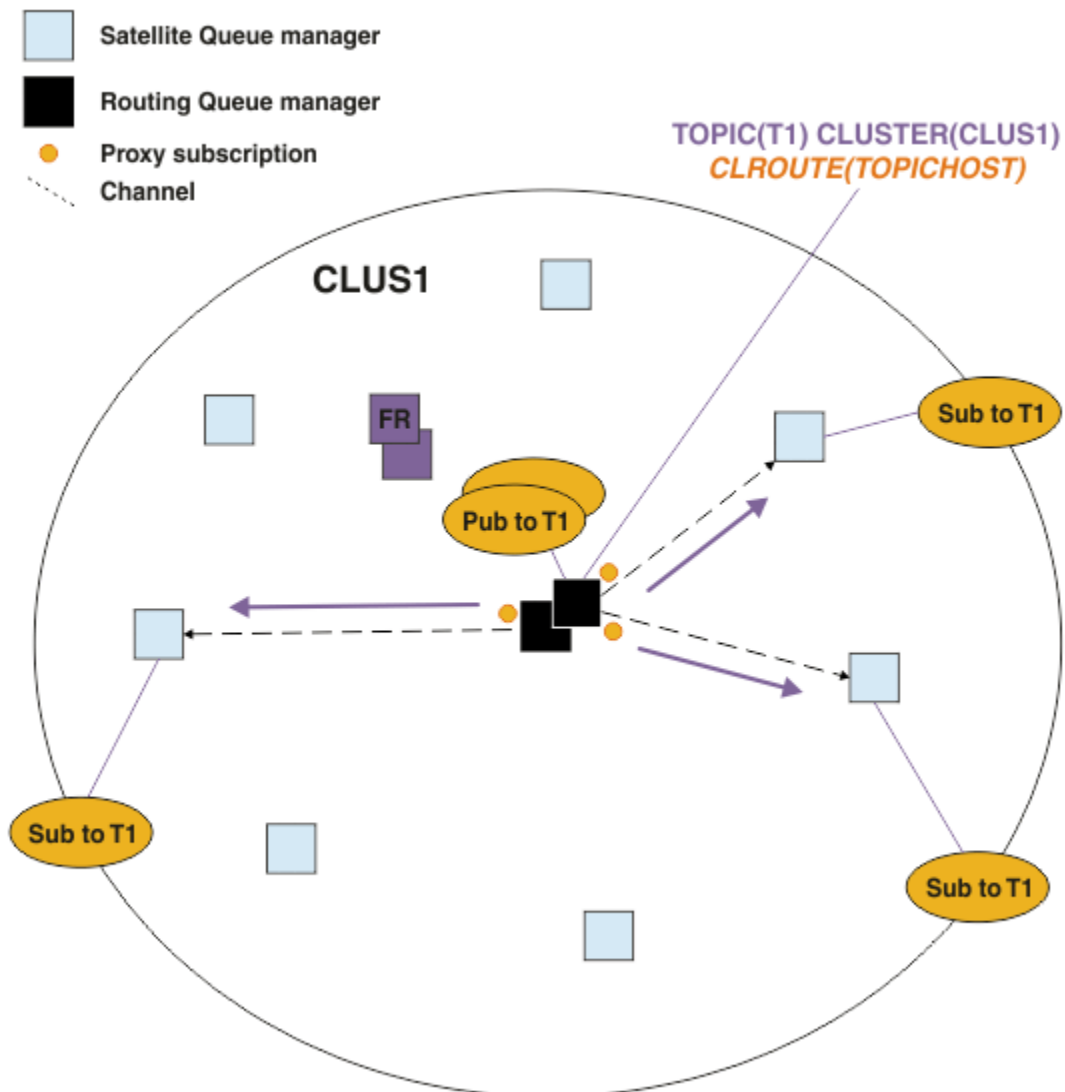


Figure 28. Hébergement de publications sur un gestionnaire de files d'attente hôte de rubrique

### Concepts associés

#### Performances du cluster de publication / abonnement par routage direct

Dans les clusters de publication / abonnement routés directement, les informations telles que les rubriques de cluster et les abonnements de proxy sont envoyées à tous les membres du cluster, que tous les gestionnaires de files d'attente de cluster participent activement à la messagerie de publication / abonnement. Ce processus peut créer une charge supplémentaire importante sur le système. Pour réduire l'impact de la gestion de cluster sur les performances, vous pouvez effectuer des mises à jour aux heures creuses, définir un sous-ensemble beaucoup plus petit de gestionnaires de files d'attente impliqués dans la publication / l'abonnement et créer un cluster "chevauchant" ou passer à l'utilisation du routage via l'hôte de rubrique.

#### Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement

Un concept important dans les performances de messagerie asynchrone est *balance*. A moins que les consommateurs de messages ne soient équilibrés avec les expéditeurs de messages, il existe un risque qu'un arriéré de messages non consommés puisse s'accumuler et affecter sérieusement les performances de plusieurs applications.

#### Performances des abonnements dans les réseaux de publication / abonnement

La publication / l'abonnement distribué dans IBM MQ fonctionne en propageant la connaissance de l'emplacement où des abonnements à des chaînes de rubrique différentes ont été créés dans le réseau du gestionnaire de files d'attente. Cela permet au gestionnaire de files d'attente sur lequel un message

est publié d'identifier les autres gestionnaires de files d'attente qui nécessitent une copie du message publié pour correspondre à leurs abonnements.

## Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement

Un concept important dans les performances de messagerie asynchrone est *balance*. A moins que les consommateurs de messages ne soient équilibrés avec les expéditeurs de messages, il existe un risque qu'un arriéré de messages non consommés puisse s'accumuler et affecter sérieusement les performances de plusieurs applications.

Dans une topologie de messagerie point-à-point, la relation entre les consommateurs de message et les expéditeurs de message est facilement comprise. Vous pouvez obtenir des estimations de la production et de la consommation de messages, file d'attente par file d'attente, canal par canal. S'il y a un manque d'équilibre, les goulets d'étranglement sont facilement identifiés, puis résolus.

Il est plus difficile de déterminer si les diffuseurs de publications et les abonnés sont équilibrés dans une topologie de publication / abonnement. Commencez à partir de chaque abonnement et revenez aux gestionnaires de files d'attente ayant des diffuseurs sur la rubrique. Calculez le nombre de publications transmises à chaque abonné à partir de chaque gestionnaire de files d'attente.

Chaque publication correspondant à un abonnement sur un gestionnaire de files d'attente éloignées (basé sur des abonnements de proxy) est placée dans une file d'attente de transmission. Si plusieurs gestionnaires de files d'attente éloignées ont des abonnements de proxy pour cette publication, plusieurs copies du message sont placées dans une file d'attente de transmission, chacune étant ciblée pour un canal émetteur différent.

Dans un cluster de publication / abonnement, ces publications sont ciblées sur la file d'attente SYSTEM . INTER . QMGR . PUBS des gestionnaires de files d'attente éloignées qui hébergent les abonnements. Dans une hiérarchie, chaque publication est destinée à la file d'attente SYSTEM . BROKER . DEFAULT . STREAM ou à toute autre file d'attente de flux répertoriée dans SYSTEM . QPUBSUB . QUEUE . NAMELIST sur les gestionnaires de files d'attente éloignées. Chaque gestionnaire de files d'attente traite les messages qui arrivent dans cette file d'attente et les distribue aux abonnements appropriés sur ce gestionnaire de files d'attente.

Pour cette raison, surveillez la charge aux points suivants où des goulets d'étranglement peuvent se produire:

- Surveillez la charge au niveau des files d'attente d'abonnement individuelles.
  - Ce goulot d'étranglement implique que l'application d'abonnement ne consomme pas les publications aussi rapidement qu'elles sont publiées.
- Surveillez la charge dans la file d'attente SYSTEM . INTER . QMGR . PUBS ou dans les files d'attente de flux.
  - Ce goulot d'étranglement implique que le gestionnaire de files d'attente reçoit les publications d'un ou de plusieurs gestionnaires de files d'attente éloignées plus rapidement qu'il ne peut les distribuer aux abonnements locaux.
  - Lorsqu'il est visible sur un gestionnaire de files d'attente d'hôte de rubrique lors de l'utilisation du routage d'hôte de rubrique dans un cluster, envisagez de créer des hôtes de rubriques de gestionnaires de files d'attente supplémentaires, ce qui permet d'équilibrer la charge de travail de publication entre eux. Toutefois, cela aura une incidence sur l'ordre des messages dans les publications. Voir [Routage d'hôte de rubrique à l'aide de plusieurs hôtes de rubrique pour une seule rubrique](#).
- Surveillez la charge sur les canaux entre le gestionnaire de files d'attente de publication et les gestionnaires de files d'attente d'abonnement, qui sont alimentés par les files d'attente de transmission sur le gestionnaire de files d'attente de publication.
  - Ce goulot d'étranglement implique qu'un ou plusieurs canaux ne sont pas en cours d'exécution ou que des messages sont publiés sur le gestionnaire de files d'attente local plus rapidement que les canaux ne peuvent les distribuer sur le gestionnaire de files d'attente éloignées.

- Lorsque vous utilisez un cluster de publication / abonnement, envisagez de définir des canaux récepteurs de cluster supplémentaires sur le gestionnaire de files d'attente cible. Cela permet d'équilibrer la charge de travail de publication entre eux. Cependant, cela affecte l'ordre des messages dans les publications. Envisagez également de passer à une configuration de file d'attente de transmission de cluster multiple, car cela peut améliorer les performances dans certaines circonstances.
- Si l'application de publication utilise une interface de publication / abonnement en file d'attente, surveillez la charge sur (a) la file d'attente SYSTEM.BROKER.DEFAULT.STREAM et toute autre file d'attente de flux répertoriée dans le SYSTEM.QPUBSUB.QUEUE.NAMELIST ; et (b) la file d'attente SYSTEM.BROKER.DEFAULT.SUBPOINT et toute autre file d'attente de sous-point répertoriée dans SYSTEM.QPUBSUB.SUBPOINT.NAMELIST .
- Ce goulot d'étranglement implique que les messages sont insérés par les applications de publication locales plus rapidement que le gestionnaire de files d'attente local ne peut traiter les messages.

### Concepts associés

#### Performances du cluster de publication / abonnement par routage direct

Dans les clusters de publication / abonnement routés directement, les informations telles que les rubriques de cluster et les abonnements de proxy sont envoyées à tous les membres du cluster, que tous les gestionnaires de files d'attente de cluster participent activement à la messagerie de publication / abonnement. Ce processus peut créer une charge supplémentaire importante sur le système. Pour réduire l'impact de la gestion de cluster sur les performances, vous pouvez effectuer des mises à jour aux heures creuses, définir un sous-ensemble beaucoup plus petit de gestionnaires de files d'attente impliqués dans la publication / l'abonnement et créer un cluster "chevauchant" ou passer à l'utilisation du routage via l'hôte de rubrique.

#### Performances du cluster de publication / abonnement de l'hôte de rubriques

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. Ces hôtes de rubrique deviennent les gestionnaires de files d'attente de *routage* pour cette branche de l'arborescence de rubriques. De plus, les gestionnaires de files d'attente sans abonnements ou diffuseurs de publications n'ont pas besoin de se connecter aux hôtes de rubrique. Cette configuration peut réduire de manière significative le nombre de connexions entre les gestionnaires de files d'attente dans le cluster et la quantité d'informations transmises entre les gestionnaires de files d'attente.

#### Performances des abonnements dans les réseaux de publication / abonnement

La publication / l'abonnement distribué dans IBM MQ fonctionne en propageant la connaissance de l'emplacement où des abonnements à des chaînes de rubrique différentes ont été créés dans le réseau du gestionnaire de files d'attente. Cela permet au gestionnaire de files d'attente sur lequel un message est publié d'identifier les autres gestionnaires de files d'attente qui nécessitent une copie du message publié pour correspondre à leurs abonnements.

#### «Surveillance des clusters», à la page 325

Dans un cluster, vous pouvez surveiller les messages d'application, les messages de contrôle et les journaux. Il existe des considérations de surveillance spéciales lorsque le cluster équilibre la charge entre deux ou plusieurs instances d'une file d'attente.

## Performances des abonnements dans les réseaux de publication / abonnement

La publication / l'abonnement distribué dans IBM MQ fonctionne en propageant la connaissance de l'emplacement où des abonnements à des chaînes de rubrique différentes ont été créés dans le réseau du gestionnaire de files d'attente. Cela permet au gestionnaire de files d'attente sur lequel un message est publié d'identifier les autres gestionnaires de files d'attente qui nécessitent une copie du message publié pour correspondre à leurs abonnements.

Cette approche minimise l'envoi de messages publiés aux gestionnaires de files d'attente sur lesquels il n'existe aucun abonnement correspondant. Cependant, la propagation de la connaissance de l'abonnement peut devenir une surcharge importante, lorsque le nombre de chaînes de rubrique

auxquelles l'abonnement est abonné est élevé et change constamment par la création et la suppression fréquentes d'abonnements.

Vous pouvez affecter les performances en ajustant la façon dont les publications et les abonnements sont transmis autour de votre réseau de publication / abonnement. Si le trafic de votre réseau comporte peu de publications et une création, une suppression ou une modification rapide des abonnements, vous pouvez arrêter de transmettre les informations d'abonnement à tous les gestionnaires de files d'attente et, à la place, transférer toutes les publications à tous les gestionnaires de files d'attente du réseau. Vous pouvez également restreindre le flux des abonnements de proxy et des publications pour une rubrique donnée entre les gestionnaires de files d'attente connectés, restreindre le flux des abonnements de proxy contenant des caractères génériques et réduire le nombre et la nature transitoire des chaînes de rubrique.

## **Propagation d'abonnements individuels et *publication partout***

*Publier partout* est une alternative à la propagation d'abonnement individuel. Avec la propagation individuelle, seules les publications ayant un abonnement correspondant sur un gestionnaire de files d'attente sont réacheminées vers ce gestionnaire de files d'attente. Avec la *publication partout*, toutes les publications sont réacheminées vers tous les gestionnaires de files d'attente du réseau. Les gestionnaires de files d'attente de réception distribuent ensuite les publications qui correspondent aux abonnements locaux.

### **Propagation d'abonnement individuel**

Ce mécanisme permet de limiter le trafic des publications entre gestionnaires de files d'attente, car seules les publications correspondant aux abonnements d'un gestionnaire de files d'attente sont envoyées.

Toutefois :

- Pour chaque chaîne de rubrique individuelle à laquelle est abonné, un abonnement de proxy est envoyé à d'autres gestionnaires de files d'attente dans la topologie de publication / abonnement. L'ensemble des gestionnaires de files d'attente dépend du modèle de routage utilisé, comme décrit dans [Planification de votre réseau de publication / abonnement distribué](#).
  - Cette surcharge de messagerie peut être importante s'il existe plusieurs milliers d'abonnements à créer ou à supprimer (par exemple, recréation de tous les abonnements non durables après un redémarrage d'un gestionnaire de files d'attente), ou si l'ensemble d'abonnements change rapidement et que chacun est associé à une chaîne de rubrique différente.
  - Le nombre de gestionnaires de files d'attente auxquels l'abonnement de proxy est propagé affecte également l'échelle du temps système.
- Les abonnements de proxy sont transmis à d'autres gestionnaires de files d'attente à l'aide de la messagerie asynchrone. Cette opération a les effets suivants :
  - Il existe un délai entre la création d'un abonnement et la création, la distribution et le traitement de l'abonnement proxy par les autres gestionnaires de files d'attente.
  - Les messages publiés sur ces gestionnaires de files d'attente dans cet intervalle ne sont pas distribués à l'abonnement distant.

### **Publier partout**

Avec ce mécanisme, il n'y a pas de surcharge par abonnement de proxy de chaîne de rubrique sur le système. Cela signifie que la création, la suppression ou la modification rapide d'un abonnement n'entraîne pas une augmentation de la charge et du traitement du réseau.

Il n'y a pas non plus de délai entre la création d'un abonnement et le flux de publications vers un gestionnaire de files d'attente, car toutes les publications sont destinées à tous les gestionnaires de files d'attente. Par conséquent, il n'y a pas de fenêtre dans laquelle les publications ne sont pas distribuées aux abonnements distants nouvellement créés.

Toutefois :

- L'envoi de toutes les publications à tous les gestionnaires de files d'attente de la topologie de publication / abonnement peut entraîner un trafic réseau excessif dans lequel les publications n'ont pas d'abonnements correspondants sur chaque gestionnaire de files d'attente.

- Plus le nombre de gestionnaires de files d'attente dans la topologie est élevé, plus le temps système est élevé.

Vous devez envisager d'utiliser le mécanisme de *publication partout* lorsque vous vous attendez à ce qu'une publication soit abonnée à partir d'une proportion importante de vos gestionnaires de files d'attente ou lorsque les frais généraux d'abonnement du proxy sont trop importants en raison de la fréquence des changements d'abonnement. Vous devez utiliser le transfert d'abonnement de proxy individuel dans les cas où vous rencontrez un trafic de messagerie accru lorsque des publications sont envoyées à tous les gestionnaires de files d'attente, plutôt qu'aux gestionnaires de files d'attente avec des abonnements correspondants.

Vous pouvez définir le comportement de *publication partout* à n'importe quel niveau de l'arborescence de rubriques. 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. Lorsqu'il est défini sur un objet de rubrique en cluster, l'attribut **PROXYSUB (FORCE)** est propagé à chaque gestionnaire de files d'attente du réseau, et pas seulement au gestionnaire de files d'attente sur lequel la rubrique a été définie.

**Remarque :** Lorsqu'il est utilisé dans une hiérarchie, vous définissez **PROXYSUB (FORCE)** individuellement sur chaque gestionnaire de files d'attente, de sorte que le mécanisme de topologie limite naturellement le nombre de canaux. Toutefois, lorsqu'il est utilisé dans un cluster, de nombreux canaux supplémentaires peuvent être démarrés:

- Dans un cluster routé par un hôte de rubrique, les canaux sont démarrés de chaque gestionnaire de files d'attente vers chaque gestionnaire de files d'attente hôte de rubrique.
- Dans un cluster routé directement, les canaux sont démarrés de chaque gestionnaire de files d'attente vers tous les autres gestionnaires de files d'attente.

Le temps système de démarrage de nombreux canaux est plus prononcé dans un cluster routé directement et peut entraîner des problèmes de performances. Voir [«Performances du cluster de publication / abonnement par routage direct»](#), à la page 389.

## Autres moyens de restreindre le flux des abonnements de proxy et des publications entre les gestionnaires de files d'attente connectés

### Consolider les chaînes de rubrique

L'utilisation de nombreuses chaînes de rubrique distinctes, transitoires, introduit un certain niveau de surcharge de gestion sur chaque gestionnaire de files d'attente du système sur lequel les diffuseurs de publications ou les abonnements sont connectés. Vous devez évaluer périodiquement l'utilisation des chaînes de rubrique pour voir si elles peuvent être consolidées. La réduction du nombre et de la nature transitoire des chaînes de rubrique, et donc des diffuseurs de publications et des abonnements associés, réduit l'impact sur le système.

### Restreindre la portée de publication et d'abonnement

Pour une rubrique donnée, vous pouvez utiliser les paramètres [Portée de la publication](#) et [Portée de l'abonnement](#) pour conserver les publications et les abonnements en local par rapport au gestionnaire de files d'attente sur lequel ils sont définis.

### Bloquer les abonnements aux rubriques génériques

Vous pouvez restreindre le flux des abonnements de proxy contenant des caractères génériques en définissant l'attribut **Topic WILDCARD** sur BLOCK. Voir [Caractères génériques dans les abonnements de proxy](#).

Voir aussi [«Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement»](#), à la page 395

## Surveillance du trafic d'abonnement de proxy dans les clusters

Lorsque vous prenez en compte la charge sur le système à partir du trafic d'abonnement du proxy, en plus de surveiller les files d'attente répertoriées dans [«Équilibrage des producteurs et des consommateurs](#)

dans les réseaux de publication / abonnement», à la page 395, surveillez les files d'attente de cluster suivantes:

- SYSTEM.INTER.QMGR.FANREQ sur le gestionnaire de files d'attente d'abonné.
- SYSTEM.INTER.QMGR.CONTROL sur tous les autres gestionnaires de files d'attente du cluster.

Tout journal des messages en attente important dans ces files d'attente implique que le taux de modification de l'abonnement est trop élevé pour le système ou qu'un gestionnaire de files d'attente ne fonctionne pas correctement dans le cluster. Si vous pensez que le problème provient d'un gestionnaire de files d'attente spécifique, vérifiez que la prise en charge de la publication / abonnement n'est pas désactivée pour ce gestionnaire de files d'attente. Voir **PSMODE** dans [ALTER QMGR](#).

### Concepts associés

#### [Performances du cluster de publication / abonnement par routage direct](#)

Dans les clusters de publication / abonnement routés directement, les informations telles que les rubriques de cluster et les abonnements de proxy sont envoyées à tous les membres du cluster, que tous les gestionnaires de files d'attente de cluster participent activement à la messagerie de publication / abonnement. Ce processus peut créer une charge supplémentaire importante sur le système. Pour réduire l'impact de la gestion de cluster sur les performances, vous pouvez effectuer des mises à jour aux heures creuses, définir un sous-ensemble beaucoup plus petit de gestionnaires de files d'attente impliqués dans la publication / l'abonnement et créer un cluster "chevauchant" ou passer à l'utilisation du routage via l'hôte de rubrique.

#### [Performances du cluster de publication / abonnement de l'hôte de rubriques](#)

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. Ces hôtes de rubrique deviennent les gestionnaires de files d'attente de *routage* pour cette branche de l'arborescence de rubriques. De plus, les gestionnaires de files d'attente sans abonnements ou diffuseurs de publications n'ont pas besoin de se connecter aux hôtes de rubrique. Cette configuration peut réduire de manière significative le nombre de connexions entre les gestionnaires de files d'attente dans le cluster et la quantité d'informations transmises entre les gestionnaires de files d'attente.

#### [Équilibrage des producteurs et des consommateurs dans les réseaux de publication / abonnement](#)

Un concept important dans les performances de messagerie asynchrone est *balance*. A moins que les consommateurs de messages ne soient équilibrés avec les expéditeurs de messages, il existe un risque qu'un arriéré de messages non consommés puisse s'accumuler et affecter sérieusement les performances de plusieurs applications.

#### [Abonnements de proxy dans un réseau de publication/abonnement](#)

## Réduction du nombre de rubriques non souhaitées dans l'arborescence de rubrique

Les performances d'un système de publication / abonnement sont améliorées en réduisant le nombre de rubriques indésirables dans l'arborescence de rubriques. Qu'est-ce qu'un sujet indésirable et comment les supprimer?

Vous pouvez créer un grand nombre de rubriques sans affecter les performances. Toutefois, certaines méthodes d'utilisation de la publication / abonnement permettent d'étendre en permanence les arborescences de rubriques. Un nombre exceptionnellement élevé de sujets sont créés une fois et ne sont plus jamais utilisés. Le nombre croissant de sujets peut devenir un problème de performances.

Comment éviter les conceptions qui conduisent à un nombre important et croissant de sujets indésirables? Que pouvez-vous faire pour aider le gestionnaire de files d'attente à supprimer les rubriques indésirables de l'arborescence de rubriques?

Le gestionnaire de files d'attente reconnaît une rubrique indésirable car elle est inutilisée depuis 30 minutes. Le gestionnaire de files d'attente supprime les rubriques inutilisées de l'arborescence de rubriques pour vous. La durée de 30 minutes peut être modifiée en modifiant l'attribut de gestionnaire de files d'attente, **TREELIFE**. Vous pouvez aider le gestionnaire de files d'attente à supprimer les rubriques inutiles en vous assurant que la rubrique apparaît dans le gestionnaire de files d'attente comme étant

inutilisée. La section [«Qu'est-ce qu'un sujet non utilisé?»](#), à la page 400 explique ce qu'est une rubrique inutilisée.

Un programmeur, la conception d'une application, et surtout la conception d'une application à exécution longue, tient compte de son utilisation des ressources: combien de ressources le programme a besoin, y a-t-il des demandes non limitées et des fuites de ressources? Les rubriques sont une ressource utilisée par les programmes de publication / abonnement. Scruter l'utilisation des sujets comme n'importe quelle autre ressource utilisée par un programme.

## Qu'est-ce qu'un sujet non utilisé?

Avant de définir ce qu'est une rubrique inutilisée, qu'est-ce qui compte exactement comme une rubrique?

Lorsqu'une chaîne de rubrique, telle que USA/Alabama/Auburn, est convertie en rubrique, la rubrique est ajoutée à l'arborescence de rubriques. Des noeuds de rubrique supplémentaires et les rubriques correspondantes sont créés dans l'arborescence, si nécessaire. La chaîne de rubrique USA/Alabama/Auburn est convertie en une arborescence comportant trois rubriques.

- USA
- USA/Alabama
- USA/Alabama/Auburn

Pour afficher toutes les rubriques de l'arborescence de rubriques, utilisez la commande **runmqsc** DISPLAY TPSTATUS(' #') TYPE(TOPIC).

Une rubrique inutilisée dans l'arborescence de rubriques possède les propriétés suivantes.

### Il n'est pas associé à un objet de rubrique

Un objet de rubrique d'administration possède une chaîne de rubrique qui l'associe à une rubrique. Lorsque vous définissez l'objet de rubrique Alabama, si la rubrique USA/Alabama à laquelle il doit être associé n'existe pas, la rubrique est créée à partir de la chaîne de rubrique. Si la rubrique existe, l'objet de rubrique et la rubrique sont associés à l'aide de la chaîne de rubrique.

### Il n'a pas de publication conservée

Une rubrique avec une publication conservée résulte d'un diffuseur de publications qui place un message dans une rubrique avec l'option MQPMO\_RETAIN.

Utilisez la commande **runmqsc** DISPLAY TPSTATUS(' USA/Alabama ') RETAINED pour vérifier si USA/Alabama possède une publication conservée. La réponse est YES ou NO.

Utilisez la commande **runmqsc** CLEAR TOPICSTR(' USA/Alabama ') CLTRTYPE(RETAINED) pour supprimer une publication conservée de USA/Alabama.

### Il n'a pas de rubriques enfant

USA/Alabama/Auburn est une rubrique sans rubrique enfant. USA/Alabama/Auburn est la rubrique enfant directe de USA/Alabama.

Affichez les enfants directs de USA/Alabama à l'aide de la commande **runmqsc** DISPLAY TPSTATUS(' USA/Alabama/+ ').

### Il n'y a aucun diffuseur actif sur le noeud

Un diffuseur de publications actif sur un noeud est une application dont la rubrique est ouverte pour la sortie.

Par exemple, une application ouvre l'objet de rubrique nommé **Alabama** avec les options d'ouverture MQOO\_OUTPUT.

Pour afficher les diffuseurs de publications actifs dans USA/Alabama et tous ses enfants, utilisez la commande **runmqsc** DISPLAY TPSTATUS(' USA/Alabama/#') TYPE(PUB) ACTCONN.



## Il n'y a pas d'abonnés actifs au noeud

Un abonné actif peut être un abonnement durable ou une application qui a enregistré un abonnement à une rubrique auprès de MQSUBet qui ne l'a pas fermé.

Pour afficher les abonnements actifs à USA/Alabama, utilisez la commande **runmqsc** DISPLAY TPSTATUS('USA/Alabama') TYPE(SUB) ACTCONN.

Pour afficher les abonnements actifs à USA/Alabama et à tous ses enfants, utilisez la commande **runmqsc** DISPLAY TPSTATUS('USA/Alabama/#') TYPE(SUB) ACTCONN.

## Gestion du nombre de rubriques dans une arborescence de rubriques

En résumé, il existe plusieurs façons de gérer le nombre de rubriques dans une arborescence de rubriques.

### Affichage TPCOUNT

Utilisez la **runmqsc** commande DISPLAY PUBSUB ALL régulièrement pour afficher la propriété **TPCOUNT** . Il s'agit du nombre de noeuds de rubrique dans l'arborescence de rubriques. Si le nombre augmente, cela peut indiquer qu'un TREELIFE plus court est requis ou qu'une nouvelle conception des rubriques elles-mêmes est requise.

### Modifiez TREELIFE

Une rubrique inutilisée a une durée de vie de 30 minutes par défaut. Vous pouvez réduire la durée de vie d'une rubrique inutilisée.

Par exemple, la commande **runmqsc** , ALTER QMGR TREELIFE(900), réduit la durée de vie d'une rubrique inutilisée de 30 minutes à 15 minutes.

### Exceptionnellement, redémarrez le gestionnaire de files d'attente

Lorsque le gestionnaire de files d'attente est redémarré, l'arborescence de rubriques est réinitialisée à partir d'objets de rubrique, de noeuds avec des publications conservées et d'abonnements durables. Les rubriques qui avaient été créées par le fonctionnement des programmes d'éditeur et d'abonné sont supprimées.

En dernier ressort, si la croissance des sujets indésirables a été la cause de problèmes de performances dans le passé, redémarrez le gestionnaire de files d'attente.

### Concepts associés

[Arborescence de rubriques](#)

## **Aspera gateway peut améliorer les performances sur les réseaux à haute latence**

IBM Aspera faspio Gateway fournit un tunnel TCP/IP rapide qui peut augmenter considérablement le débit du réseau pour IBM MQ.

Aspera gateway peut être utilisé pour améliorer les performances des canaux de gestionnaire de files d'attente. Elle est particulièrement efficace si le réseau a un temps d'attente élevé ou a tendance à perdre des paquets, et elle est généralement utilisée pour accélérer la connexion entre les gestionnaires de files d'attente dans différents centres de données.

Cependant, pour un réseau rapide qui ne perd pas de paquets, il y a une diminution des performances lors de l'utilisation de Aspera gateway, il est donc important de vérifier les performances du réseau avant et après la définition d'une connexion Aspera gateway .

Un gestionnaire de files d'attente s'exécutant sur une plateforme autorisée peut se connecter via un Aspera gateway. La passerelle elle-même est déployée sur Red Hat<sup>®</sup> , Ubuntu Linux ou Windows.

Pour plus d'informations, voir [Définition d'une connexion Aspera gateway sous Linux ou Windows](#).



## 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: